Chapter 2
Complex Networks

Abstract Complex network comprises an emerging interdisciplinary research area that triggers much attention from physicists, mathematicians, biologists, engineering, computer scientists, among many others. Complex network structures describe a wide variety of systems of high technological and intellectual importance, such as the Internet, World Wide Web, coupled biological and chemical systems, financial, social, neural, and communication networks. The desire to understand such interwoven systems summed with their inherent complexity are factors that explain the increasing interest in enhancing complex network tools. The data representation in complex networks permits us to unify the structural complexity and vertex and connection diversities. Several relevant questions arise when investigating dynamics in complex networks, such as learning how large ensembles of dynamical systems that interact through a complex wiring topology can behave collectively. In this way, the network topology plays an important role in that it affects the functions of the represented system. As an example, the structure of social networks affects the information and disease propagation speeds, the topology of a financial network may amplify shocks in different manners, and the disposition of power grids in networks may affect the robustness and stability of power transmission. Due to the rapid evolution and the large amount of developed theories and techniques, it becomes prohibitive to make a comprehensive review on this topic. In this chapter, we present the basic concepts and ideas of complex networks that are useful in machine learning. We start out by presenting the main concepts of networks. Since complex networks and graphs share the same definition, we first present the basic notations of graph theory. Afterwards, we explore the evolution line and milestones of the complex network research. Following that, a comprehensive list of network measurements is discussed, which enables us to capture structural features of the networks in a systematic manner. Finally, we present some well-known dynamical processes that are defined within the complex networks framework.

2.1 Basic Concepts of Graphs

In this section, we discuss fundamental concepts of the graph theory.

T.C. Silva, L. Zhao, *Machine Learning in Complex Networks*,
DOI 10.1007/978-3-319-17290-3_2

2.1.1 Graph Definitions

We here present the main terminology employed by the literature of graphs or networks theory. In this book, the words graphs and networks convey the same type of information and are used interchangeably. In the same spirit, the data relationships that make up a graph are termed structure, topology, or anatomy of the network.

In the following, we present the formal definition of a graph [8, 21, 35].

Definition 2.1. Graph: A graph $\mathscr{G}$ is defined as an ordered pair $\langle \mathscr{V}, \mathscr{E} \rangle$, where $\mathscr{V}$ is a finite nonempty set of vertices or nodes and $\mathscr{E}$ is the set of edges or links between the vertices $\mathscr{E} \subseteq \{(u, v) \mid u, v \in \mathscr{V}\}$. Some special graphs are defined as follows:

- **Graph with no self-loops**: When the relation $\mathscr{E}$ is irreflexive, meaning that $\forall v \in \mathscr{V}, (v, v) \notin \mathscr{E}$, the graph is said to be free of self-loops. This means that there is no way of traveling to the same vertex in a single transition.
- **Graph with self-loops**: When the relation $\mathscr{E}$ satisfies the following restriction $\exists v \in \mathscr{V}, (v, v) \in \mathscr{E}$, the graph is said to have self-loops. This means that one can travel back to the same vertex through an edge without leaving it.

Moreover, we denote by $V = |\mathscr{V}|$ and $E = |\mathscr{E}|$ the number of vertices and edges, respectively, of the graph.

For example, in the graph $\mathscr{G}$ portrayed in Fig. 2.1, the vertex set is $\mathscr{V} = \{1, \ldots, 5\}$ and the edge set is $\mathscr{E} = \{(1, 2), (1, 3), (2, 3), (3, 3), (3, 4)\}$. We often label the edges with letters or numbers. In the same example, another possible edge labeling is $\mathscr{E} = \{e_1, e_2, e_3, e_4, e_5\}$, where $e_1 = (1, 2)$, $e_2 = (1, 3)$, $e_3 = (2, 3)$, $e_4 = (3, 3)$, $e_5 = (3, 4)$. We can check from Definition 2.1 that the existence of the edge $e_4 = (v_3, v_3)$ turns $\mathscr{G}$ into a graph with self-loops.

Some well-known graph topologies are discussed in the following.

Definition 2.2. Complete graph: A complete graph is a graph in which links exist between each pair of vertices. The complete graph with V vertices is denoted by $\mathscr{K}_V$.

A complete graph can also be further classified into with or without self-loops, in accordance with Definition 2.1.

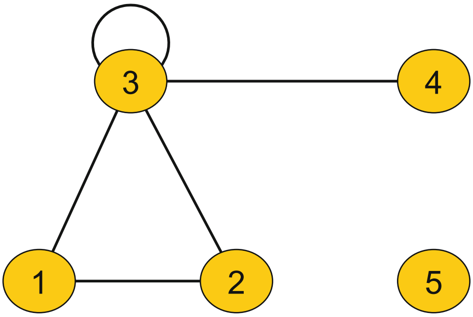

Fig. 2.1 An example of graph that is undirected, non-weighted, and with self-loops

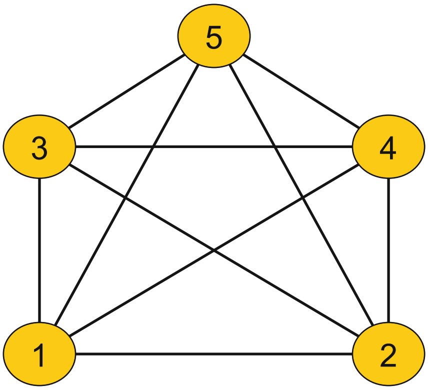

Fig. 2.2 An example of complete graph that is non-weighted and with no self-loops

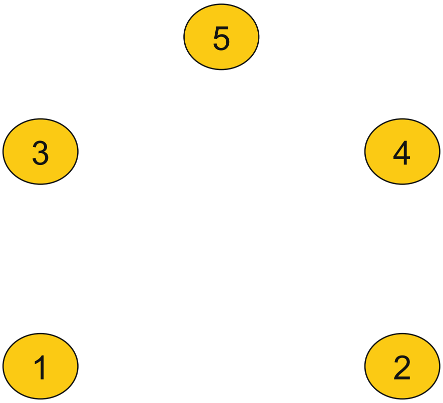

Fig. 2.3 An example of null graph

For example, Fig. 2.2 shows a complete graph $\mathscr{K}_5$, where every pair of vertices is connected by an edge. As there are no self-loops, the graph is considered a complete graph with no self-loops.

Definition 2.3. Null graph: A null graph is a graph containing no edges, that is, $\mathscr{E} = \emptyset$.

Figure 2.3 illustrates a null graph with 5 vertices. We highlight that, even though the edge set is empty, that is, $\mathscr{E} = \emptyset$, the vertex set cannot be empty. Otherwise, we would not have a formal graph in view of Definition 2.1.

Graphs can also be classified with respect to their edge types. In the next, we discuss the main edge types encountered in the literature.

Definition 2.4. Undirected graph: When the relation $\mathscr{E}$ is symmetric, meaning that $\forall (u, v) \in \mathscr{E} \Rightarrow (v, u) \in \mathscr{E}$, it is said that the graph is undirected. In other terms, when there is an edge linking vertices u to v, so there will be a link from v to u.

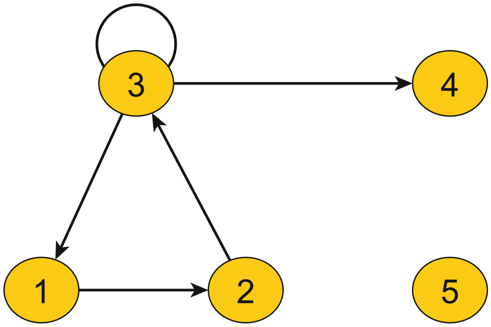

Fig. 2.4 An example of a graph that is directed (digraph), non-weighted, and with self-loops

In the illustrative graph in Fig. 2.1, we note that besides the edge set $\mathscr{E}_1 = \{(1, 2), (1, 3), (2, 3), (3, 3), (3, 4)\}$, the set $\mathscr{E}_2 = \{(2, 1), (3, 1), (3, 2), (3, 3), (4, 3)\}$ is also present. Therefore, the graph in Fig. 2.1 is undirected.

Commonly, when the edge $(u, v) \in \mathscr{E}$ is drawn with no arrows in its endpoints, it is assumed that the edge is undirected, implying the existence of the opposite edge $(v, u) \in \mathscr{E}$. This pictorial distinction is made clear with the definition of directed graphs in the following.

Definition 2.5. Directed graph (digraph): When the relation $\mathscr{E}$ satisfies the following restriction: $\exists(u, v) \in \mathscr{E} \mid (v, u) \notin \mathscr{E}$, it is said that the graph is directed (digraph). In other terms, this kind of graph must have at least an arbitrary edge linking u to v, with an absence of the opposite link.

Figure 2.4 gives an example of a directed graph. In this case, each edge has its direction, which is conveyed by the visual illustration of the graph itself. The directness of the graph implies that there exists at least one edge $(u, v) \in \mathscr{E}$ such that $(v, u) \notin \mathscr{E}$. This holds true in Fig. 2.4 for several cases. Among them, we can see that the edge $(1, 2) \in \mathscr{E}$, but $(2, 1) \notin \mathscr{E}$.

There is a special type of graph known as weighted graph, whose definition is given as follows. The same graph categories discussed in Definition 2.1 can be applied to it [13, 32].

Definition 2.6. Weighted graph: A weighted graph $\mathscr{G}$ is defined as a triple $\mathscr{G} = \langle \mathscr{V}, \mathscr{E}, \mathbf{W} \rangle$, where $\mathscr{V}$ and $\mathscr{E}$ are the sets of vertices and edges, respectively, and $\mathbf{W}$ is a matrix that carries the edge weights. For example, the entry $\mathbf{W}_{uv} = w$, $(u, v) \in \mathscr{E}$, fixes as $w > 0$ the weight of the edge linking vertices u to v. If $(u, v) \notin \mathscr{E} \Rightarrow \mathbf{W}_{uv} = 0$.

Figure 2.5 shows an example of a weighted graph where each edge is associated to a value. Often, when no edge weight is specified, it is assumed that the weight is unitary.

The weights can convey various types of meanings in different applications. For example, each value (weight) may represent the distance from vertex (location) i to j, or it may also represent traffic flow and so on. By setting the edge weight with large or small values, we are effectively adjusting the importance of that edge for the

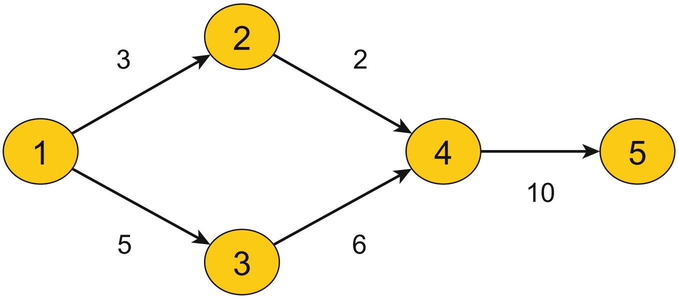

Fig. 2.5 An example of a graph that is weighted, directed (digraph), and with no self-loops

application that we are dealing with. For instance, in graph-based machine learning, each weight frequently represents the similarity degree between two vertices (data samples). As such, large values denote a close proximity of those vertices and, hence, a high importance is given to that relationship in the learning process.

Remark 2.1. When $\mathbf{W}$ is a binary matrix, then the weighted graph reduces to a non-weighted graph, which is the special graph supplied in Definition 2.1.

Definition 2.7. Bipartite graph: A bipartite graph is a graph whose set of vertices $\mathscr{V}$ can be split into two disjoint non-empty subsets $\mathscr{V}_1$ and $\mathscr{V}_2$, $\mathscr{V} = \mathscr{V}_1 \bigcup \mathscr{V}_2$, in such a way that $(u, v) \in \mathscr{E} \Rightarrow u \in \mathscr{V}_1, v \in \mathscr{V}_2$. Therefore, no edge exists between pairs of vertices in the same subsets $\mathscr{V}_1$ and $\mathscr{V}_2$.

Remark 2.2. Note that, if $\mathscr{G}$ is a bipartite graph, then $\mathscr{G}$ cannot have self-loops.

Remark 2.3. We say that $\mathscr{G}$ is a complete bipartite graph $\mathscr{K}_{M,N}$ when $|\mathscr{V}_1| = M$ and $|\mathscr{V}_2| = N$ and $\forall (v, u) \in \mathscr{V}_1 \times \mathscr{V}_2, (v, u) \in \mathscr{E}$.

When modeling relations between two different classes of objects, bipartite graphs very often arise naturally. Some examples are:

- The graph of football players and clubs, in which an edge exists between a player and a club if that player has played for that club, is a natural example of an affiliation network, a type of bipartite graph used in social network analysis.
- The graph that represents job allocation in a company. A boss must allocate in a company N open jobs for M workers. Each worker is qualified to do some of the N jobs, but not others. Links will exist between a worker and his/her specified qualified jobs.

Figure 2.6 depicts a bipartite graph with $V = 5$ vertices, where $\mathscr{V}_1 = \{1, 2, 3\}$ and $\mathscr{V}_2 = \{4, 5\}$. Note that the existence of links only occurs between vertices of $\mathscr{V}_1$ and $\mathscr{V}_2$.

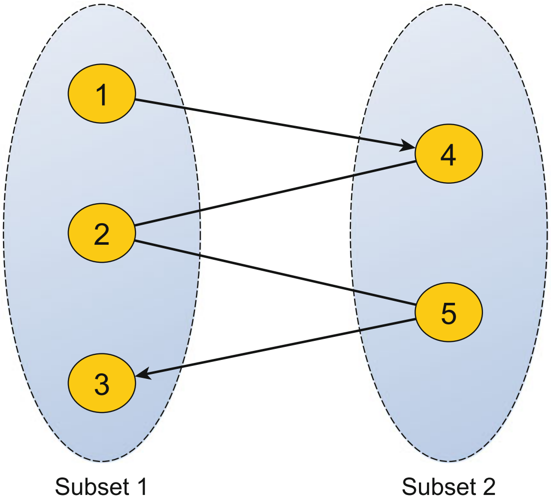

Fig. 2.6 An example of a graph that is bipartite, non-weighted, and directed (digraph)

2.1.2 Connectivity

In this section, we introduce common terms related to graph connectivity, which are used throughout this book [8, 13, 21, 32, 35].

Definition 2.8. Adjacent vertices: Two vertices $u \in \mathscr{V}$ and $v \in \mathscr{V}$ are called adjacent if they share a common edge, in which case the common edge is said to join the two vertices.

Remark 2.4. In undirected graphs, if u is adjacent to v, then v must be adjacent to u as well.

Remark 2.5. In digraphs, u adjacent to v does not imply that v is adjacent to u. Specifically, if $(u, v) \in \mathscr{E}$ and $(v, u) \notin \mathscr{E}$, then v is adjacent to u, but the opposite does not hold.

For instance, in the undirected graph portrayed in Fig. 2.7a, vertices 1 and 3 are adjacent to each other. In contrast, vertex 1 is not adjacent to 4. Now, in the directed graph depicted in Fig. 2.7b, vertex 1 is adjacent to 3, but the converse is not true.

Definition 2.9. Neighborhood of a vertex: The neighborhood of a vertex $v \in \mathscr{V}$, in a graph $\mathscr{G}$ is the set of vertices adjacent to v. The neighborhood is denoted by $\mathscr{N}(v)$ and is formally given by $\mathscr{N}(v) = \{u : (v, u) \in \mathscr{E}\}$.

For illustrative purposes, in the undirected graph shown in Fig. 2.7a, the neighborhood of vertex 1 is $\mathscr{N}(1) = \{2, 3\}$. Now, in the directed graph exhibited in Fig. 2.7b, $\mathscr{N}(1) = \{2\}$.

Remark 2.6. Some authors further distinguish the neighborhood of a vertex in open and closed neighborhoods. The *open neighborhood* of v never includes v itself. The

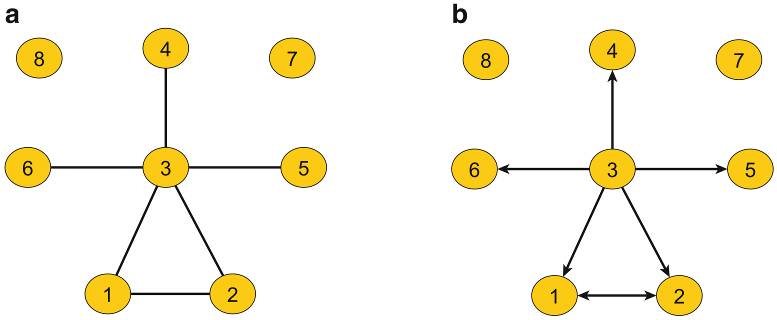

Fig. 2.7 Illustrative non-weighted graphs for exemplifying basic graph concepts. (**a**) Undirected graph. (**b**) Directed graph

closed neighborhood extends the previous one by adding v itself into $\mathscr{N}(v)$, i.e., $\mathscr{N}^{(\text{closed})}(v) = \mathscr{N}(v) \bigcup \{v\}$. In this book, we opt not to discriminate between these classes of neighborhood, because in some machine learning algorithms, self-loops are allowed to prevent transition to other vertices. Therefore, if $v \in \mathscr{N}(v) \iff (v, v) \in \mathscr{E}$. That is, the condition of v being neighbor to itself only depends on the existence of a self-loop in v. We find that this notation is more intuitive and consistent with the machine learning literature.

Definition 2.10. Degree (valency or connectivity) of a vertex: In an undirected graph, the degree of a vertex v is the total number of vertices adjacent to v. The degree of a vertex v is denoted by k_v. We can equivalently define the degree of a vertex as the cardinality of its neighborhood set and say that, for any vertex v, $k_v = |\mathscr{N}(v)|$, i.e.,

$$k_v = |\mathscr{N}(v)| = |\{u : (v, u) \in \mathscr{E}\}| = \sum_{u \in \mathscr{V}} \mathbb{1}_{[(v,u) \in \mathscr{E}]}, \tag{2.1}$$

in which $\mathbb{1}_{[K]}$ represents the Kronecker delta or indicator function that yields 1 if the logical expression K is true; otherwise, it returns 0.

Remark 2.7. The feasible values of k_v are within the discrete-valued interval $\{0, \ldots, V - 1\}$ if self-loops are not allowed, and in $\{0, \ldots, V\}$ if self-loops are permitted.

Remark 2.8. When $k_v = 0$, then v is said to be a *singleton* or *isolated* vertex.

Remark 2.9. When k_v assumes relative large values than the remainder of the vertices in the network, we say that v is a *hub*.

In the undirected graph depicted in Fig. 2.7a, vertices 7 and 8 are singleton, for $k_7 = k_8 = 0$. In contrast, vertex 3 is considered as a hub, for its degree is relative large in relation to the remainder vertices of the network.

We have so far discussed definitions mostly suited to undirected graphs. For directed graphs, some of the previously defined connectivity measures suffer slight modifications, mainly due to the fact that distinctions in the edge endpoints must be brought into consideration. In special, a directed edge has two distinct ends: an origin and a destination. The measures use these two endpoints independently. In light of these considerations, we now extend the connectivity definitions to the case of directed networks.

Definition 2.11. In-degree and out-degree: In a directed graph, the notion of vertex degree can be further extended into the in-degree, $k_v^{(\text{in})}$, and out-degree, $k_v^{(\text{out})}$, as follows:

$$k_v^{(\text{in})} = \sum_{u \in \mathscr{V}} \mathbb{1}_{[v \,\in\, \mathscr{N}(u)]} = \sum_{u \in \mathscr{V}} \mathbb{1}_{[(u,v) \,\in\, \mathscr{E}]}, \tag{2.2}$$

$$k_v^{(\text{out})} = \sum_{u \in \mathscr{V}} \mathbb{1}_{[u \,\in\, \mathscr{N}(v)]} = \sum_{u \in \mathscr{V}} \mathbb{1}_{[(v,u) \,\in\, \mathscr{E}]}, \tag{2.3}$$

$$k_v = k_v^{(\text{in})} + k_v^{(\text{out})}. \tag{2.4}$$

Remark 2.10. The domains of $k_v^{(\text{out})}$ and $k_v^{(\text{in})}$ are $\{0, \ldots, V-1\}$ if self-loops are not allowed, and $\{0, \ldots, V\}$ if self-loops are permitted. Therefore, k_v may assume the values $\{0, \ldots, 2(V-1)\}$ when no loops are present and $\{0, \ldots, 2V\}$ when loops are allowed.

Remark 2.11. Note that $k_v^{(\text{out})} = |\mathscr{N}(v)|$.

For example, in the directed graph exhibited in Fig. 2.7b, $k_3^{(\text{out})} = 5$, $k_3^{(\text{in})} = 0$, and $k_3 = 5 + 0 = 5$. In addition, $k_1^{(\text{out})} = 1$, $k_1^{(\text{in})} = 2$, and $k_1 = 1 + 2 = 3$.

Definition 2.12. Average network degree: The average degree of the network, or network connectivity, is given by:

$$\bar{k} = \frac{1}{V} \sum_{v \in \mathscr{V}} k_v = \frac{1}{V} \sum_{(v,u) \in \mathscr{V}^2} \mathbb{1}_{[(v,u) \,\in\, \mathscr{E}]}. \tag{2.5}$$

For instance, in the undirected graph exhibited in Fig. 2.7a, the average degree is:

$$\bar{k} = \frac{1}{8}\left[k_1 + \ldots + k_8\right] = \frac{1}{8}\left[2 + 2 + 5 + 1 + 1 + 1 + 0 + 0\right] = 1.5,$$

i.e., on average, a vertex belonging to that network has 1.5 links.

Definition 2.13. Average in-degree and out-degree: In a directed graph, the average in-degree and out-degree have the same numerical value and are evaluated as:

$$\bar{k}^{(\text{in})} = \bar{k}^{(\text{out})} = \frac{1}{V}\sum_{v\in\mathscr{V}} k_v^{(\text{in})} = \frac{1}{V}\sum_{v\in\mathscr{V}} k_v^{(\text{out})}. \tag{2.6}$$

In the example shown in Fig. 2.7b, the average in- and out-degree are given by:

$$\begin{aligned}
\bar{k}^{(\text{in})} &= \frac{1}{8}\left[k_1^{(\text{in})} + \ldots + k_8^{(\text{in})}\right] &&= \frac{1}{8}\left[2+2+0+1+1+1+0+0\right] = \frac{7}{8},\\
\bar{k}^{(\text{out})} &= \frac{1}{8}\left[k_1^{(\text{out})} + \ldots + k_8^{(\text{out})}\right] &&= \frac{1}{8}\left[1+1+5+0+0+0+0+0\right] = \frac{7}{8}.
\end{aligned}$$

In the rest of this section, we define some connectivity measurements that are useful for weighted graphs.

Definition 2.14. Strength: In an undirected weighted graph, the strength of a vertex $v \in \mathscr{V}$, indicated by s_v, represents the total sum of weighted connections of v towards its neighbors.

$$s_v = \sum_{u\in\mathscr{V}} \mathbf{W}_{vu}, \tag{2.7}$$

in which $\mathbf{W}_{vu}$ is the edge weight of v to u, as introduced in Definition 2.6.

In the graph exhibited in Fig. 2.8a, $s_1 = 3 + 2 = 5$, and $s_2 = 3 + 5 + 10 = 18$.

Definition 2.15. In-strength and out-strength: In a directed weighted graph, the notion of vertex strength can be further extended into the in-strength, $s_v^{(\text{in})}$, and out-strength, $s_v^{(\text{out})}$, as follows:

$$s_v^{(\text{in})} = \sum_{u\in\mathscr{V}} \mathbf{W}_{uv}, \tag{2.8}$$

$$s_v^{(\text{out})} = \sum_{u\in\mathscr{V}} \mathbf{W}_{vu}, \tag{2.9}$$

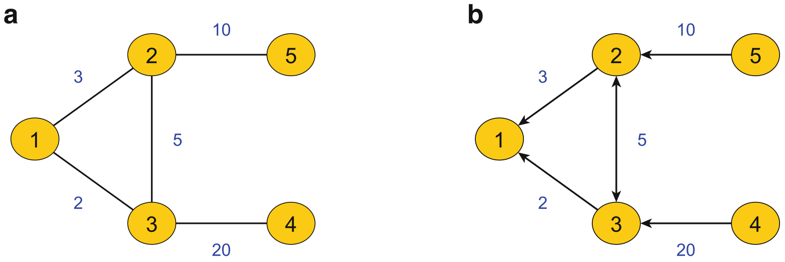

Fig. 2.8 Illustrative weighted graphs for exemplifying basic graph concepts. (**a**) Undirected graph. (**b**) Directed graph

$$s_v = s_v^{(\mathrm{in})} + s_v^{(\mathrm{out})}, \tag{2.10}$$

in which $\mathbf{W}_{vu}$ is the edge weight linking v to u.

In the example supplied in Fig. 2.8b, for vertex 1, we have $s_1^{(\mathrm{in})} = 3 + 2 = 5$ and $s_1^{(\mathrm{out})} = 0$. Similarly, for vertex 2, $s_2^{(\mathrm{in})} = 5 + 10 = 15$ and $s_2^{(\mathrm{out})} = 3 + 5 = 8$.

With the basic connectivity concepts introduced, we present another well-known graph topology in the following.

Definition 2.16. **Regular graph**: A graph is regular if all of the graph vertices have the same degree. In particular, if the degree of each vertex is k, $\mathscr{G}$ is said to be k-regular.

Remark 2.12. If $\mathscr{G}$ is a complete graph with V vertices, then it is $(V-1)$-regular. An example is the complete graph in Fig. 2.2, which is 4-regular with 5 vertices.

Examples of regular graphs that are not complete are supplied in Fig. 2.9. In special, Fig. 2.9a has six vertices and is a 2-regular network, while Fig. 2.9b has ten vertices and is a 3-regular network.

2.1.3 Paths and Cycles

Definition 2.17. **Walk**: Let $v_1, \ldots, v_K \in \mathscr{V}$, $K \geq 2$. A walk $\mathscr{W}$ is an ordered sequence of edges: $\mathscr{W} = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{K-1}, v_K)\}$, such that $\forall k \in \{2, \ldots, K\} : (v_{k-1}, v_k) \in \mathscr{E}$. In this case, v_1 and v_k are called the walk's origin and destination, respectively. Note that vertices can be revisited in the same walk.

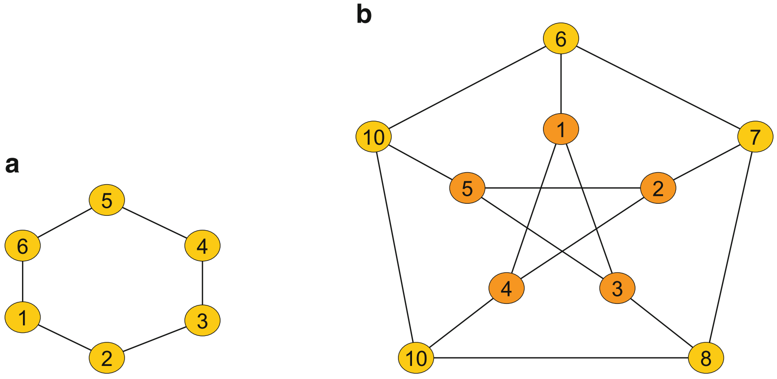

Fig. 2.9 Examples of regular graphs that are not complete. (**a**) 2-regular graph. (**b**) 3-regular graph (Petersen graph)

Remark 2.13. A walk is called *closed* if $v_1 = v_K$ and *open* otherwise.

Remark 2.14. A walk consisting of a single vertex is called a *trivial* walk.

Definition 2.18. Trail: A trail is a walk in which no edge is repeated. Trails can also be further classified into open and closed trails, according to Remark 2.13.

Definition 2.19. Tour or circuit: A tour is a closed trail.

Definition 2.20. Walk length: The length of a walk $\mathscr{W} = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{K-1}, v_K)\}$, $K \geq 2$, is the number of edges that the walk traverses, i.e., $|\mathscr{W}| = K - 1 \geq 1$.

In the undirected graph portrayed in Fig. 2.10, $\mathscr{W}_1 = \{(1,3), (3,4), (4,6), (6,7), (7,4)\}$ is an open walk. In contrast, $\mathscr{W}_2 = \{(1,3), (3,4), (4,6), (6,7), (7,4), (4,3), (3,1)\}$ is a closed walk. There are no trivial walks, as the graph in Fig. 2.10 has no self-loops. $\mathscr{W}_3 = \{(5,8), (8,7)\}$ is an open trail and $\mathscr{W}_4 = \{(5,8), (8,7), (7,5)\}$ is a closed trail or a tour. The lengths of these walks are: $|\mathscr{W}_1| = 5$, $|\mathscr{W}_2| = 7$, $|\mathscr{W}_3| = 2$, and $|\mathscr{W}_4| = 3$. There are no walks that visit vertex 10.

Definition 2.21. Path: A path $\mathscr{P}$ is a non-trivial walk in which all vertices (except possibly the first and last) are distinct.

Remark 2.15. A path is always a walk.

Definition 2.22. Cycle: A cycle is closed path.

In Fig. 2.10, $\mathscr{P}_1 = \{(1,2), (2,5), (5,7)\}$ is a path and $\mathscr{P}_2 = \{(1,2), (2,5), (5,7), (7,4), (4,3), (3,1)\}$ is a cycle. Note that $\mathscr{P}_3 = \{(5,8), (8,7), (7,6), (6,4), (4,7), (7,5), (5,8)\}$ is a walk and tour but not a cycle, because it is not even a path.

Definition 2.23. Walk or path distance: The distance d of the walk $\mathscr{W} = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{K-1}, v_K)\}$, $K \geq 2$ is given by:

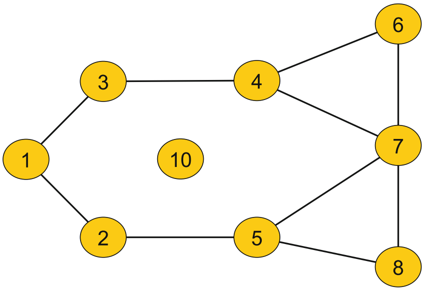

Fig. 2.10 Illustrative undirected graph to introduce graph traversal measures

$$d(\mathscr{W}) = \sum_{k=2}^{K} |(v_{k-1}, v_k)| = \sum_{k=2}^{K} \mathbf{W}_{k-1,k}, \tag{2.11}$$

in which $|(v_{k-1}, v_k)|$ is the edge weight linking vertex v_{k-1} to v_k.

Definition 2.24. Shortest path (geodesic path) between vertices: The shortest path between $u \in \mathscr{V}$ and $v \in \mathscr{V}$, denoted here as d_{uv}, is given by the path starting from u and ending at v with the least distance. Mathematically,

$$d_{uv} = \min_{\mathscr{W}_{u \to v}} d(\mathscr{W}_{u \to v}), \tag{2.12}$$

in which $\mathscr{W}_{u \to v}$ represents walks starting from u and ending at v.

Remark 2.16. For measures that require two inputs, such as the shortest path between vertices, we use d_{uv} when the subscripts are variables and $d_{1,2}$ when they are numbers. That is, we maintain the notation as succinct as possible. The comma is employed for clarity when numbers are indexed.

Definition 2.25. Distance between vertices: The distance d_{uv} between two vertices u and v is always their shortest path distance.

Remark 2.17. Note that d_{uv} is always evaluated from a path. That is, the distance between u and v cannot be a walk that is not a path.

Remark 2.18. The distance between any vertex and itself is 0.

Remark 2.19. If there is no path from u to v, then $d_{uv} = \infty$.

In Fig. 2.10, the distance between 1 and 3 is $d_{1,3} = 1$, since the shortest path from 1 to 3 is $\{(1,3)\}$. The distance from vertex 10 to itself is $d_{10,10} = 0$. Moreover, the distance from vertex 1 to 10 is $d_{1,10} = \infty$, as no paths nor walks exist between 1 and 10.

2.1.4 Subgraphs

Definition 2.26. Reachability: We say that $v_2 \in \mathscr{V}$ is reachable from $v_1 \in \mathscr{V}$ if $d_{v_1 v_2}$ is finite. Alternatively, v_1 reaches v_2 if there is at least a walk that starts from v_1 and ends at v_2.

Definition 2.27. Connectedness: Graph $\mathscr{G}$ is connected if, for every pair of vertices v_1 and v_2, v_2 is reachable from v_1 *or* v_1 is reachable from v_2.

Definition 2.28. Strong connectedness: Graph $\mathscr{G}$ is strongly connected if, for every pair of vertices v_1 and v_2, v_2 is reachable from v_1 *and* v_1 is reachable from v_2.

Remark 2.20. Strong connectedness implies connectedness.

Remark 2.21. In undirected graphs, connectedness implies strong connectedness. This holds true because if v_1 reaches v_2, then the converse must be true, for edges are two-way in undirected graphs.

Remark 2.22. In directed graphs, connectedness does not imply strong connectedness.

In the undirected graph depicted in Fig. 2.11a, the graph is strongly connected and, hence, each pair of vertices is mutually reachable. In contrast, in the directed graph exhibited in Fig. 2.11b, the graph is connected but not strongly connected. For instance, v_1 reaches v_6 but the converse is not true. The graphs in Fig. 2.11c, d are not strongly connected nor connected. For example, v_1 and v_8 are mutually non-reachable.

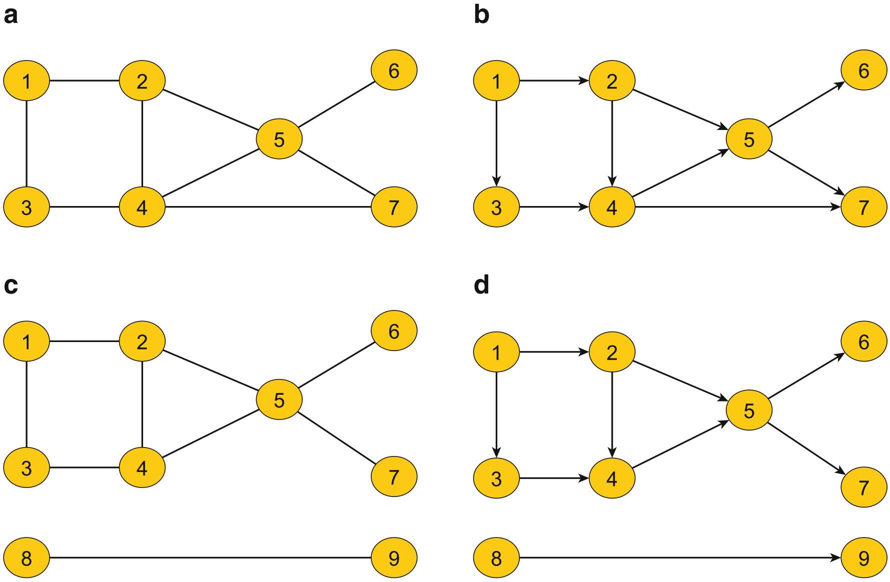

Fig. 2.11 Illustrative graphs for exemplifying subgraph concepts. (**a**) Undirected graph (1 component). (**b**) Directed graph (1 component). (**c**) Undirected graph (2 components). (**d**) Directed graph (2 components)

Definition 2.29. Graph component: The subgraph $\mathscr{G}_C$ of $\mathscr{G}$ is a component if:

- $\mathscr{G}_C$ is connected;
- All of the proper subsets of $\mathscr{G}_C$ are not connected.

Alternatively, $\mathscr{G}_C$ is a graph component if any two of its vertices are reachable at least from one to another, and if its vertex members are connected to no additional vertices in the remainder of the graph.

Remark 2.23. A connected graph has always a single component.

In Fig. 2.11a, b, there is a single component that is the graph itself. In contrast, in Fig. 2.11c, d, two components exist: $\mathscr{G}_1 = \{1, 2, 3, 4, 5, 6, 7\}$ and $\mathscr{G}_2 = \{8, 9\}$.

Definition 2.30. Clique: A clique in an undirected graph is a subset of vertices such that every two vertices in the subset are connected by an edge. Cliques therefore are subgraphs or graphs that are complete.

In Fig. 2.11a, there are two cliques: one comprises the vertices $\{4, 5, 7\}$, while the other, $\{2, 4, 5\}$.

2.1.5 Trees and Forest

Definition 2.31. Tree graph: A tree is a connected graph that has no cycles. In a tree, a leaf is a vertex of degree 1. An internal vertex is a vertex of degree at least 2.

Definition 2.32. Forest: A forest is an undirected graph in which all of its connected components are trees.

Remark 2.24. Note that a forest is a graph consisting of a disjoint union of trees.

Remark 2.25. All trees are forests, but the converse is not always true.

Remark 2.26. Special cases of forests include: a single tree and a graph with only singleton vertices (empty graph).

Figure 2.12a illustrates a tree, while Fig. 2.12b exemplifies a forest with two trees.

Definition 2.33. Spanning tree: If $\mathscr{G}$ is a connected graph, the spanning tree in $\mathscr{G}$ is a subgraph of $\mathscr{G}$ which includes every vertex of $\mathscr{G}$ and is also a tree graph.

For example, Fig. 2.13b shows a possible spanning tree from the graph exhibited in Fig. 2.13a. In this transformation process, we have removed the edges $(2, 3), (2, 4), (3, 5), (4, 5), (6, 7)$.

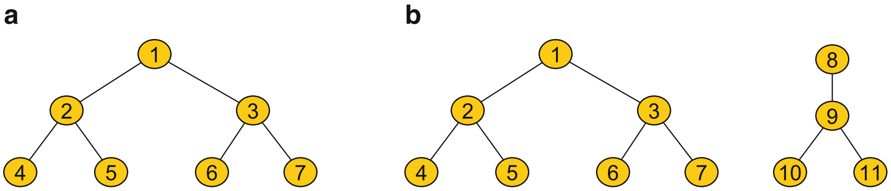

Fig. 2.12 Examples of special types of graphs: trees and forests. (**a**) A tree. (**b**) A forest with two trees

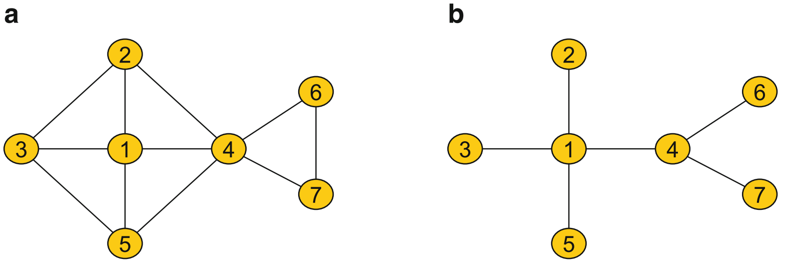

Fig. 2.13 Transformation of a graph in (**a**) into a spanning tree in (**b**)

2.1.6 Graph Representation

Mathematically, a non-weighted graph $\mathscr{G} = \langle \mathscr{V}, \mathscr{E} \rangle$ or weighted graph $\mathscr{G} = \langle \mathscr{V}, \mathscr{E}, \mathbf{W} \rangle$ are frequently represented by an adjacency matrix **A** that is constructed from the vertex and edge sets. A formal definition of the adjacency matrix is given as follows.

Definition 2.34. Adjacency matrix: Let $\mathscr{G} = \langle \mathscr{V}, \mathscr{E}, \mathbf{W} \rangle$ be an weighted graph. Then, the adjacency matrix **A** is defined as follows:

- The number of vertices $|\mathscr{V}| = V$ serves to establish the dimension of the adjacency matrix, which is always $V \times V$;
- The edge set contributes to defining the entry values of the adjacency matrix in the following manner. The (i, j)-th entry of **A** is denoted as $\mathbf{A}_{ij} = a_{ij} = \mathbf{W}_{ij}$, where $\mathbf{W}_{ij}$ is the weight of the edge linking i to j. Formally, $\forall (i, j) \in \mathscr{E} : a_{ij} \neq 0$ and $\forall (i, j) \notin \mathscr{E} : a_{ij} = 0$.

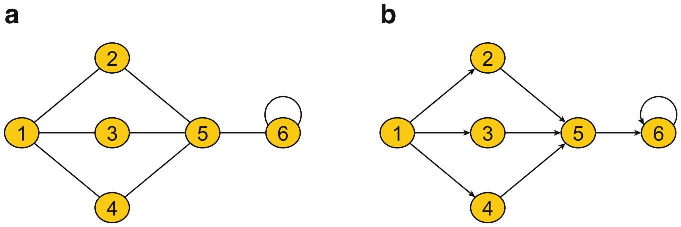

Fig. 2.14 Illustrative graphs introduced for evaluating their adjacency matrices. (**a**) Undirected graph. (**b**) Directed graph

Usually, the adjacency matrix **A** takes the following matrix form:

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,V} \\ a_{2,1} & a_{2,2} & \dots & a_{2,V} \\ \vdots & \vdots & \ddots & \vdots \\ a_{V,1} & a_{V,2} & \dots & a_{V,V} \end{pmatrix}. \tag{2.13}$$

Remark 2.27. If $\mathscr{G}$ is non-weighted, then $\mathbf{A}_{ij} \in \{0, 1\}$, $\forall i, j \in \mathscr{V}$.

Remark 2.28. If the graph $\mathscr{G}$ is undirected, then **A** is symmetric. This fact implies that if $\mathbf{A}_{ij} = 1$, then $\mathbf{A}_{ji} = 1$.

Remark 2.29. Contrasting to the previous Remark, directed graphs may not have symmetric adjacency matrices, as j can be a neighbor of i and the converse may not hold.

For instance, the undirected graph shown in Fig. 2.14a has the following adjacency matrix:

$$\mathbf{A} = \mathbf{A}^T = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{2.14}$$

in which the superscript T denotes the transpose operator.

In addition, note that the matrix in (2.14) is symmetric, i.e., $\mathbf{A} = \mathbf{A}^T$, as the graph in Fig. 2.14a is undirected.

In contrast, the directed graph exhibited in Fig. 2.14b has the following adjacency matrix:

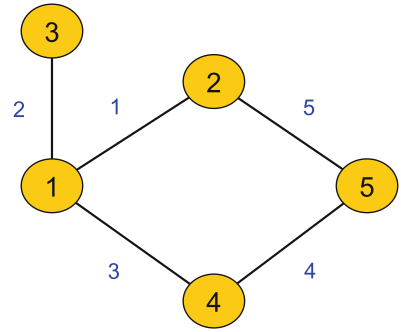

Fig. 2.15 Illustrative weighted graph introduced for evaluating the weighted matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.15}$$

In this case, the matrix in (2.15) is not symmetric.

For weighted graphs, the entries in the adjacency matrix can assume arbitrary values. For instance, the (weighted) adjacency matrix of the weighted undirected graph portrayed in Fig. 2.15 is:

$$\mathbf{A} = \mathbf{A}^T = \begin{pmatrix} 0 & 1 & 2 & 3 & 0 \\ 1 & 0 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 4 \\ 0 & 5 & 0 & 4 & 0 \end{pmatrix}. \tag{2.16}$$

2.2 Complex Network Models

With the expectation of studying topological properties that are linked to real networks, several network models have been proposed. Some of these models even have inspired an extensive study due to its features of great interest. As examples of important categories of networks, one can list: random networks, small-world networks, clustered random networks, scale-free networks, and core-periphery networks. In the next sections, we review these models in detail.

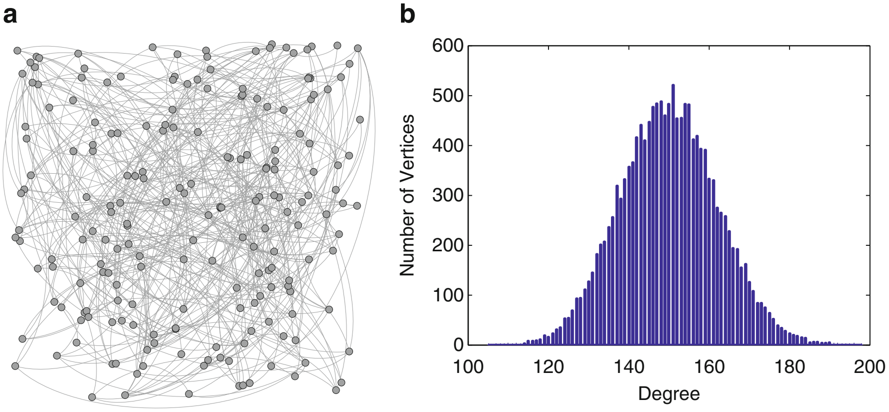

Fig. 2.16 An example of random networks of Erdös and Réyni. (**a**) A network constructed by means of the random approach proposed by Erdös e Réyni; and (**b**) the degree distribution of a network consisting of $V = 15,000$ constructed using the Erdös and Réyni methodology with $p = 0.01$

2.2.1 Random Networks

In the article dated back to 1959, Erdös and Réyni [24] developed a model that generates random networks consisting of V vertices and E edges. Starting from V vertices completely disconnected (no edges in the network), the network is built from the gradual addition of L edges randomly created, in such a way that self-looping is avoided. Another similar model sets V vertices in a network, and there is a probability $p > 0$ of connecting each possible pair of vertices. The latter model is widely recognized as the model of Erdös and Réyni. Figure 2.16a depicts an example of this type of network. Note that no spacial relation between the vertices is used. In this network formation, we merely create edges in a uniform probabilistic way, regardless of the similarity between vertices.

Since, for each vertex $i \in \mathcal{V}$ of the network (a total of V), there are $V-1$ different possibilities of connections with other vertices, it follows that the cardinality of the sample space, $|\Omega|$, which quantifies the maximum theoretical number of edges between the vertices, is given by:

$$|\Omega| = \frac{V(V-1)}{2}, \tag{2.17}$$

in which the division by 2 comes from the fact that we are considering that the graph is undirected, i.e., the edges are always bidirectional in relation to both linked vertices. In general, the presence of these two edges represents the occurrence of the same probabilistic event, on account of the inherent coupling (bidirectionally). Having in mind that an arbitrary edge is present in a random network with probability p and is absent with probability $1-p$, and remembering that there are

$\binom{V-1}{k}$ ways of choosing k vertices over $V-1$ in total, and p^k denotes the joint probability of these k vertices to possess exactly k connected vertices,[1] then $\binom{V-1}{k}p^k$ provides the probability of these k vertices to have exactly k other interconnected vertices. However, in this analysis, it should be imposed that there are no more edges beyond these k, i.e., for the reminiscent quantity of vertices, $V-1-k$, the complementary probabilistic event of existing edges, that is, $(1-p)^{(V-1-k)}$, must happen. In view of this reasoning, the degree distribution follows a Binomial distribution with parameters $Binomial(V-1, p)$, whose equation is governed by the following expression:

$$P(k) = \binom{V-1}{k} p^k (1-p)^{(V-1)-k}. \tag{2.18}$$

Given that $V \rightarrow \infty$ and $p \ll 1$, one can show that a Binomial distribution parameterized with $Binomial(V-1, p)$ asymptotically approximates a Poisson distribution with parameter $Poisson(\lambda)$ [52], with the following linking condition:

$$(V-1)p = \lambda. \tag{2.19}$$

Recall from the probability theory that the mean, μ, and the variance, σ^2, of a $Poisson(\lambda)$ are given by $\mu = \sigma^2 = \lambda$. If we construct an artificial random network using the discussed methodology with $V = 15,000$ e $p = 0.01$, we get the degree distribution that is displayed in Fig. 2.16b. Note that the resulting degree distribution really approximates the Poisson distribution with mean (peak) around $\lambda = (V-1)p = (15,000-1)0.01 \approx 150$.

Moreover, the average shortest path $\langle d \rangle$ is small in random networks. This quantity increases proportionally to the logarithm of the network size, i.e., $\langle d \rangle \sim \frac{\ln(V)}{\ln(\langle k \rangle)}$, where $\langle k \rangle$ is given by the average value of the Poisson distribution (mean degree), meaning that $\langle k \rangle = \lambda = (V-1)p$, [20].

The big discovery of Erdös and Réyni was that many important properties of a random network may be unveiled as one modifies the parameters of a $Binomial(V-1, p)$. In their study, they showed that, for values of the connecting probability p larger than a critical probability p_c, almost all of the random networks present a specific property Q with probability 1. That same property is not verified whenever $p \leq p_c$. For example, if p is larger than a certain value of p_c, the random networks can present a single connected component. But, for values below this critical threshold, the random networks no longer present a single component, but instead several unconnected subgraphs. Many other interesting properties have been discussed in the literature and some of them are reviewed in [55].

[1] The joint probability is evaluated taking into account that the existence or absence of links are independent from each other in the random network model.

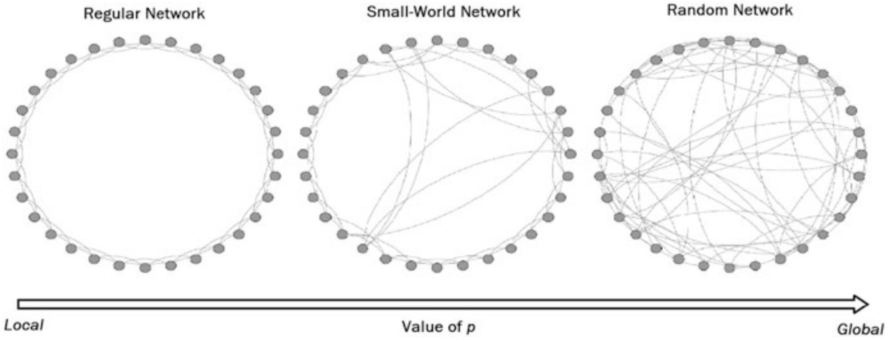

Fig. 2.17 Network behavior as we increase the parameter p, which is responsible for the relocation frequency of the edges

2.2.2 Small-World Networks

Several real-world networks exhibit the small-world property, i.e., most vertices can be reached by others by means of a small number of intermediate steps (edges). This characteristic is found, for example, in social networks, where virtually everyone in the world can be reached by a short chain of people [73, 74].

In order to build a network that presents the small-world property, one can use the following network formation process introduced in [74]:

- Initially, the network is regular, comprising V vertices, as shown in the left-most network in Fig. 2.17, in which each vertex connects to its k nearest neighbors in each direction, totalizing $2k$ connections;
- Then, each edge is randomly relocated, i.e., given an arbitrary vertex $i \in \mathcal{V}$, we randomly choose one of its original $2k$ connections. The selected edge, say linking vertices i and $j \in \mathcal{V}$, is randomly relocated, such that the destination from j is switched to another vertex $u \in \mathcal{V}, j \neq u$, with probability p.

When $p = 0$, no rearrangements are performed and, therefore, the network continues to be regular. Conversely, when $p \to 1$, all of the edges are effectively relocated [74]. Figure 2.17 brings a schematic of the behavior of the parameter p, responsible for the relocation frequency of the edges. Note that, for $p = 0$, the resulting network is virtually a regular one. As p increases (but still remains small), the property of small-world becomes apparent. When $p = 1$, the network turns out to be random. In this case, the peak of the degree distribution, following this approach, is situated close to $2k$ [73, 74].

The immediate implication for networks that have the property of small world is that the spread of any information, given that it was generated at any arbitrary vertex of the network, is very fast. For example, in viral contagion networks with the small-world property, given that a person has contracted some virus, then it is expected that, in a short time, many people will be infected by this virus due to the network topology that favors rapid propagation.

2.2.3 Scale-Free Networks

In a study conducted by Barabási and Albert [5], they noticed that some networks have a small number of vertices with large degrees, while most of them have very small degrees. With this observation in mind, in 1999, they proposed a new type of network denominated scale-free networks, in which the degree distribution obeys a power-law, as follows:

$$P(k) \sim k^{-\gamma}, \tag{2.20}$$

in which γ is a scaling exponent. Note that, by setting a fixed value for γ, as the degree k grows, the number of vertices that have degree k decreases. Thus, it is expected that $P(k)$ will have a large value for small values of k and a small value for large values of k, which is consistent with the observation found by Barabási and Albert.

The scale-free property strongly correlates with the network robustness to failure. In a scale-free network topology, it turns out that major hubs are closely followed by smaller ones. These smaller hubs, in turn, are followed by other vertices with an even smaller degree and so on until we reach peripheral or terminal vertices. This hierarchy allows for a fault tolerant behavior. If failures occur at random and the vast majority of vertices are those with small degree, the likelihood that a hub would be affected is almost negligible. Even if a hub-failure occurs, the network generally does not lose its connectedness, due to the remaining hubs. On the other hand, if we choose a few major hubs and take them out of the network, the network is turned into a set of rather isolated graphs. Thus, hubs are both a strength and a weakness of scale-free networks. In view of that, the literature often terms scale-free networks as robust to random attacks yet fragile to intentional attacks. These properties have been studied analytically using percolation theory by Cohen et al. [16, 17] and by Callaway et al. [11].

The formation of scale-free networks happens due to preferential attachment of vertices. This behavior can be understood in terms of network growth. Growth in this context means that the number of vertices in the network increases over time. Preferential attachment means that the more connected a vertex is, the more likely it is to receive new links. Vertices with larger degree have stronger ability to grab links added to the network. Intuitively, the preferential attachment can be understood if we think in terms of social networks connecting people. Here a link from A to B means that person A "knows" or "is acquainted with" person "B." Heavily linked vertices represent well-known people with lots of relations. When a newcomer enters the community, he or she is more likely to become acquainted with one of those more visible people rather than with a relative unknown. Similarly, on the web, new pages link preferentially to hubs, i.e., very well-known sites such as Google or Wikipedia, rather than to pages that hardly anyone knows. If someone selects a new page to link to by randomly choosing an existing link, the probability of selecting a particular page would be proportional to its degree. This explains the preferential attachment probability rule. Preferential attachment is an example of a positive feedback cycle where initially random variations are automatically reinforced, thus greatly magnifying differences.

Albert and Barabási [5] proposed an algorithm to generate scale-free network with this preferential attachment mechanism. The network begins with an initial connected network of V_0 vertices. New vertices are added to the network one at a time. Each new vertex is connected to $V \leq V_0$ existing vertices with a probability that is proportional to the number of links that the existing vertices already have. Formally, the probability p_i that the new vertex is connected to vertex i is:

$$p_i = \frac{k_i}{\sum_{j \in \mathcal{V}} k_j}, \tag{2.21}$$

in which k_i is the degree of vertex i. Heavily linked vertices or hubs tend to quickly accumulate even more links, while vertices with only a few links are unlikely to be chosen as the destination for a new link. The new vertices have a “preference” to attach themselves to the already heavily linked vertices.

Figure 2.18 shows an illustrative network that shares the scale-free properties. Note that there are very few vertices with large degree, while the great majority (terminal vertices) has small degree.

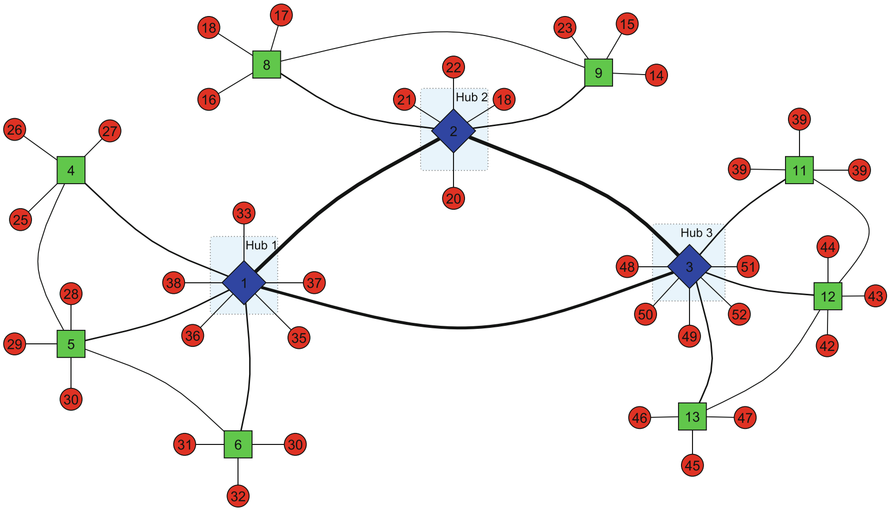

Fig. 2.18 Schematic of a scale-free network. The hubs (vertices with large degrees) have been evidenced. Note that there are very few vertices with large degrees, while the great majority (terminal vertices) has small degrees

2.2.4 Random Clustered Networks

Some real-world networks, such as social and biological ones, present modular structures called communities [31]. These communities consist of sets of vertices that satisfy a simple rule: vertices belonging to the same community have many interconnecting edges, while different communities share relatively few edges interconnecting each other. A model for generating such communities was proposed in [31]. This agglomerative method groups V initially isolated vertices into M communities. This is managed by creating a link between two vertices with probability p_{in}, if they belong to the same community, or with probability p_{out}, if they belong to distinct communities. The values for p_{in} and p_{out} can be arbitrarily chosen to control the number of intracommunity and intercommunity links, z_{in} and z_{out}, respectively, for an arbitrary average network degree $\langle k \rangle$.

High values of p_{in} and low values of p_{out} refer to networks with well-defined communities, i.e., there is a high concentration of edges confined within each community and very few edges interconnecting different communities. Conversely, low values of p_{in} and high values of p_{out} contribute to the appearance of communities highly mixed with each other. On the basis of these parameters, we can define the fraction of intracommunity links $z_{\text{in}}/\langle k \rangle$ and, likewise, the fraction of intercommunity links $z_{\text{out}}/\langle k \rangle$. The quantity $z_{\text{out}}/\langle k \rangle$ defines the mixture among different communities. Essentially, as $z_{\text{out}}/\langle k \rangle$ increases, the communities become more mixed and harder to be identified. As we will further see in Sect. 6.2.4, these quantities are usually employed to compare different competing community detection techniques using the Girvan-Newman's benchmark, which adopts the random clustered networks discussed here.

Empirically, $p_{\text{out}} \ll p_{\text{in}}$ must be satisfied in order to guarantee the presence of communities in the network. Figure 2.19 illustrates a network with four communities. Observe that the communities in this figure are well-defined, since the number of edges connecting vertices of the same community is much larger than the number of edges interconnecting those of different communities.

2.2.5 Core-Periphery Networks

Networks can be described using a combination of local, global, and intermediate-scale (mesoscale) perspectives. In this aspect, one of the key objectives of network theory is the identification of statistical summaries for large networks in order to develop frameworks that serve to analyze and compare complex structures. In such efforts, the algorithmic identification of mesoscale graph structures makes it possible to uncover features that might not be apparent neither at the local level of vertices and edges nor at the global level of statistical summaries.

In particular, several efforts have gone into the algorithmic identification and investigation of a particular type of mesoscale structure known as community

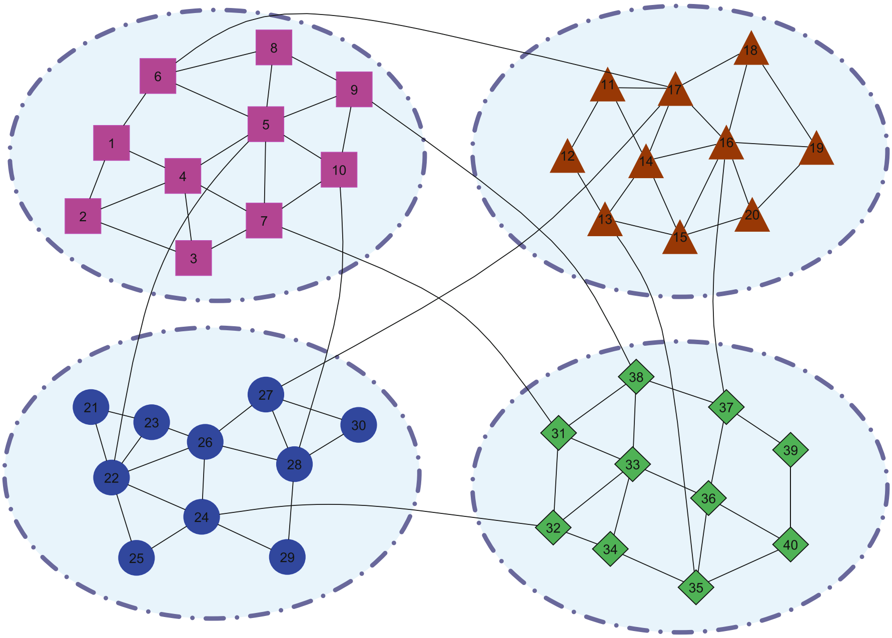

Fig. 2.19 Schematic of a random clustered network with four well-defined communities. Each community is distinguished by a unique color or format

structure, in which cohesive groups called communities consist of vertices that are densely interconnected and connections between vertices in different communities are comparatively sparse.

Although researches of community structure have been very successful [28], the investigation of other types of mesoscale structures—often in the form of different "block models" [26, 28]—have received much less attention in the literature. In this section, we deal with another kind of mesoscale network structure known as core-periphery structure. The qualitative notion that social networks can have such a structure makes intuitive sense and has a long history in subjects like sociology [22, 45], international relations [12, 70], and economics [42]. The most popular quantitative method to investigate core-periphery structure was proposed by Borgatti and Everett in 1999 [10].

By computing a network core-periphery structure, one attempts to determine which vertices are part of a densely connected core and which are part of a sparsely connected periphery. Core vertices should also be reasonably well-connected to peripheral vertices, but the latter are not well-connected to a core nor to each other. Hence, a vertex belongs to a core if and only if it is well-connected both to other core vertices and to peripheral vertices. A core structure in a network is thus not merely densely connected but also tends to be "central" to the network (e.g., in terms of short paths through the network). The goal of quantifying various notions of "centrality," which are intended to measure the importance of a vertex or other network component [58, 72], also helps in distinguishing core-periphery structure from community structure. Additionally, networks can have nested core-periphery structure as well as both core-periphery structure and community structure [46], so it is desirable to develop algorithms that allow one to simultaneously examine both types of mesoscale structure.

Hubs, which are vertices that have large degree, occur in many real-world networks and can pose a problem for community detection, as they often are connected to vertices in many parts of a network and can thus have strong ties to several different communities. For instance, such vertices might be assigned to different communities when applying different computational heuristics using the same notion of community structure [69]. Therefore, it becomes crucial to consider their strengths of membership across different communities (e.g., by using a method that allows overlapping communities) [1]. In such situations, the usual notion of a community might not be ideal for achieving an optimal understanding of the mesoscale network structure that is actually present, and considering hubs to be part of a core in a core-periphery structure might be more appropriate [46]. For example, one can consider communities as tiles that overlap to produce a network core [68, 75].

Figure 2.20 illustrates a perfect core-periphery network. We observe that core vertices are strongly interconnected to each other and also considerably connected to the remainder of the peripheral network. Peripheral vertices, in turn, are only connected to the core.

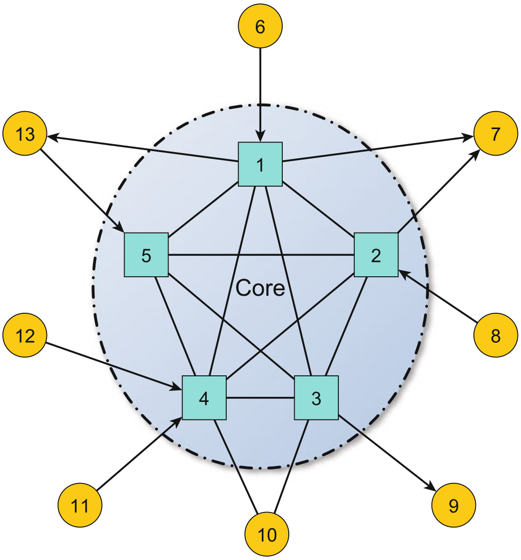

Fig. 2.20 Schematic of a core-periphery network. The core member are depicted in *square-shaped vertices* (*cyan color*) and the peripheral members, in *circle-shaped vertices* (*yellow color*)

2.3 Complex Network Measures

In this section, we review network measurements that have been proposed in the complex networks literature.

2.3.1 Degree and Degree-Correlation Measures

Definition 2.35. Density: the network density D measures how strong the vertices of a graph are connected. It is defined as the fraction of actual connections over the total possible connections.

For a directed network, the density D is defined as:

$$D = \frac{E}{2\binom{V}{2}} = \frac{2E}{2V(V-1)} = \frac{E}{V(V-1)}, \tag{2.22}$$

in which $2\binom{V}{2}$ denotes the total number of possible connections in a directed graph. In special, the binomial accounts for getting the total number of pairwise combinations between two vertices in the network. We multiply by two because the ordering (start and destination vertices) of those pairwise connections matters in a directed graph.

For an undirected network, the density D is:

$$D = \frac{E}{\binom{V}{2}} = \frac{2E}{V(V-1)}, \tag{2.23}$$

in which, in this case, the ordering of the pairwise connections does not matter.

The density assumes values in the interval $[0, 1]$. When $D = 0$, we say that $\mathscr{G}$ is an empty graph. Conversely, when $D = 1$, $\mathscr{G}$ is said to be a complete or maximal clique graph.

Remark 2.30. Often in the literature, networks can also be classified as sparse, when D assumes values near 0, and dense, otherwise. As a rule of thumb, when the number of edges in the networks is of the order of the number of vertices, i.e., $E = \mathscr{O}(V)$, the network is considered sparse. As we will see, the density of networks has profound implications on the time complexity of the majority of the machine learning algorithms. As such, it is a common practice in the literature to state the time complexity of machine learning techniques both in terms of sparse and dense networks.

Definition 2.36. Network assortativity: The network assortativity captures, in a structural sense, the preference of vertices to attach to others that are similar or different in terms of the degree [54]. Assortativity is often operationalized as a degree correlation among vertices. The assortativity coefficient r is essentially the Pearson correlation coefficient of degree between pairs of linked vertices. Hence, positive values of r indicate a correlation between vertices of similar degree, while negative values indicate relationships between vertices of different degrees [53]. In general, r lies between -1 and 1. When $r = 1$, the network is said to have perfect assortative mixing patterns, while at $r = -1$ the network is completely disassortative.

Many studies have been conducted and some conclusions have been drawn on some types of real-world networks. For example, social networks often have apparent assortative mixing. On the other hand, the technological, biological, and financial networks frequently appear to be disassortative [53].

Considering that u_e and v_e are the degrees of the two vertices at the endpoints of the e-th edge of a non-empty graph $\mathscr{G}$, and that $E = |\mathscr{E}|$ is the number of edges of $\mathscr{G}$, the network assortativity r is evaluated as follows [53]:

$$r = \frac{E^{-1}\sum_{e\in\mathscr{E}} u_e v_e - \left[\frac{E^{-1}}{2}\sum_{e\in\mathscr{E}}(u_e + v_e)\right]^2}{\frac{E^{-1}}{2}\sum_{e\in\mathscr{E}}(u_e^2 + v_e^2) - \left[\frac{E^{-1}}{2}\sum_{e\in\mathscr{E}}(u_e + v_e)\right]^2}. \tag{2.24}$$

Definition 2.37. Local assortativity: Local assortativity can be used to analyze assortative or disassortative tendencies at local level [65]. Local assortativity, denoted by r_{local}, has been defined as the individual contribution of each vertex to the network assortativity. The local assortativity of a vertex u with degree $j + 1$ is given by [65, 66]:

$$r_{\text{local}}(u) = \frac{(j+1)(j\bar{k} - \mu_q^2)}{2E\sigma_q^2}, \tag{2.25}$$

in which $\bar{k}$ is the average remaining degree of the neighbors of u, E is the number of links in the network, μ_q and σ_q are the mean and standard deviation of the remaining degree distribution of the network. It follows that the network assortativity r can be retrieved using the following expression:

$$r = \sum_{u\in\mathscr{V}} r_{\text{local}}(u). \tag{2.26}$$

Definition 2.38. Non-normalized rich-club coefficient: The rich-club coefficient first appeared in the literature as an unscaled metric parametrized by vertex degree ranks [82]. More recently, this has been updated to be parameterized in terms of vertex degrees k, indicating a degree cut-off. The rich-club coefficient measures the structural property of complex networks called "rich-club" phenomenon. This property refers to the tendency of vertices with large degree (hubs) to be tightly connected to each other, thus forming clique or near-clique structures.

This phenomenon has been discussed in several instances in both social and computer sciences. Essentially, vertices with a large number of links, usually known as rich vertices, are much more likely to form dense interconnected subgraphs (clubs) than vertices with small degree. Considering that $E_{>k}$ is the number of edges among the $N_{>k}$ vertices that have degree larger than a given threshold $k \geq 0$, the scaled version of the rich-club coefficient is expressed as [19, 51, 61]:

$$\phi(k) = \frac{2E_{>k}}{N_{>k}\,(N_{>k} - 1)}, \tag{2.27}$$

in which the factor $N_{>k}(N_{>k}-1)/2$ represents the maximum feasible number of edges that can exist among $N_{>k}$ vertices. We note that, while the network assortativity captures how connected similar vertices are in terms of degree connectivity, the rich-club coefficient can be viewed as a more specific notation of associativity, where we are only concerned with the connectivity of vertices beyond a certain richness metric. For example, if a network consists of a collection of hub and spokes, where the hubs are well connected, such a network would be considered disassortative. However, due to the strong connectedness of the hubs in the network, the network would demonstrate a strong rich-club effect.

Definition 2.39. Normalized rich-club coefficient: A criticism of the above non-normalized rich-club coefficient is that it does not necessarily imply the existence of the rich-club effect, as it is monotonically increasing even for random networks. This is true because vertices with larger degree are naturally more likely to be more densely connected than vertices with smaller degree, simply due to the fact that they have more incident edges. In fact, for certain degree distributions, it is not possible to avoid connecting hubs with large degrees. As a result, for a proper evaluation of this phenomenon, we must normalize out this factor. This point was raised in [19], which derived an analytical expression for the rich-club coefficient of uncorrelated large-size networks at large degrees. To account for this, it is necessary to compare the above metric to the same metric on a degree distribution that preserves the randomized version of the network. This updated metric is defined as [19, 51, 61]:

$$\phi_{\mathrm{norm}}(k) = \frac{\phi(k)}{\phi_{\mathrm{rand}}(k)}, \tag{2.28}$$

in which $\phi(k)$ is the non-normalized rich-club coefficient of the network under analysis and $\phi_{\mathrm{rand}}(k)$ is the non-normalized rich-club coefficient evaluated on a maximally randomized network with the same degree distribution $P(k)$ of the network under study. This new ratio discounts unavoidable structural correlations that are a result of the degree distribution, giving a better indicator of the significance of the rich-club effect. For this metric, if for certain values of k we have $\phi_{\mathrm{norm}}(k) > 1$, this denotes the presence of the rich-club effect.

Remark 2.31. Networks with strong disassortative mixing patterns that have rich-club regions composed of vertices with large degrees point for the existence of core-periphery structures (cf. Sect. 2.2.5). The number of cores, in this case, is defined as the number of graph components that results when applying the procedure to evaluate the rich-club coefficient.

2.3.2 Distance and Path Measures

Definition 2.40. Diameter: The diameter of $\mathscr{G}$, T, is the length of the largest pairwise distance in $\mathscr{G}$. Formally, it is given by:

$$T = \max_{u,v \in \mathscr{V}} d_{uv}. \tag{2.29}$$

For a non-weighted graph, the feasible values of T are $[0, V - 1]$. The diameter can be interpreted as the largest intermediation chain in the network.

Definition 2.41. Vertex eccentricity: The eccentricity of $u \in \mathscr{V}$, e_u, is the largest distance from u to any other vertex $v \in \mathscr{V}\backslash\{u\}$, i.e.:

$$e_u = \max_{v \in \mathscr{V}\backslash\{u\}} d_{uv}. \tag{2.30}$$

Definition 2.42. Radius: The network radius, ζ, is its minimum eccentricity, i.e.:

$$\zeta = \min_{u \in \mathscr{V}} e_u. \tag{2.31}$$

Definition 2.43. Wiener index: The Wiener index, λ, is defined as the sum of geodesic distances between each pair of vertices in the graph. Mathematically, it is given by:

$$\lambda = \frac{1}{2} \sum_{\substack{u,v \in \mathscr{V} \\ u \neq v}} d_{uv}. \tag{2.32}$$

One problem of this measure is its divergence for disconnected graphs, because at least one geodesic distance is infinity. Such a problem can be avoided by computing only pairs of connected vertices. However, it introduces distortion if the graph has many pairs of disconnected vertices. The following network measures, global efficiency and average harmony, are defined in a way to solve this problem.

Definition 2.44. Global efficiency: The global efficiency, GE, considers that the efficiency of sending information between two vertices u and v is inversely proportional to the geodesic distance [2], i.e.:

$$GE = \frac{1}{V\,(V-1)} \sum_{\substack{u,v \in \mathscr{V} \\ u \neq v}} \frac{1}{d_{uv}}. \tag{2.33}$$

Definition 2.45. Average Harmony: The average harmony, h, is the reciprocal of the overall global efficiency, i.e.:

$$h = \frac{1}{GE}. \tag{2.34}$$

The average harmony does not present the problem of divergence shown by Wiener index, so it is suitable for graphs with unconnected vertices [20].

2.3.3 Structural Measures

Definition 2.46. Clustering Coefficient: The clustering coefficient measure quantifies the degree to which local vertices in a network tend to cluster together. Evidence suggests that in many real-world networks, and in particular social networks, vertices tend to create tightly knit groups characterized by a relatively high density of ties [74]. Several generalizations and adaptations of such measure have been proposed in the literature [44, 60]. Here, we define the measure originally proposed by Watts and Strogatz [74]. The local clustering coefficient of a vertex in a graph quantifies how close its neighbors are to being a clique (complete graph). Mathematically speaking, the local clustering coefficient of vertex i is given by:

$$CC_i = \frac{2|e_i|}{k_i\,(k_i-1)}, \tag{2.35}$$

in which $|e_i|$ the number of links shared by the direct neighbors of vertex i (number of triangles formed by vertex i and any of its two neighbors) and k_i is the degree of vertex i. By (2.35), we see that $CC_i \in [0, 1]$.

Definition 2.47. Network clustering coefficient: We can also evaluate the network clustering coefficient, which gives us a sense of quasi-local connectivity between vertices, as follows:

$$CC = \frac{1}{V} \sum_{i \in \mathscr{V}} CC_i, \tag{2.36}$$

in which V symbolizes the number of vertices and $CC \in [0, 1]$. Roughly speaking, the clustering coefficient tells how well-connected the neighborhood of the vertex is.

If the neighborhood is fully connected, the clustering coefficient is 1 and a value close to 0 means that there are hardly any triangular connections in the neighborhood.

Definition 2.48. Cyclic Coefficient: This coefficient characterizes the degree of circulation in complex networks by considering cycles of all orders from 3 up to infinity [39]. The cyclic coefficient θ_i of vertex i is the average of the inverse size of the smallest cycle that connects that vertex and any of two of its neighbor vertices. Mathematically, it is calculated as follows [39]:

$$\theta_i = \frac{2}{k_i\,(k_i - 1)} \sum_{j,k \in \mathscr{N}(i)} \frac{1}{S^i_{jk}}, \tag{2.37}$$

in which S^i_{jk} is the smallest size of the closed shortest path that passes through vertex i and its two neighbor vertices j and k. Note that the sum goes over all of the neighbor pairs (j, k) of i. If vertices j and k are directly linked to each other, then vertices i, j, and k form a triangle. It is a cycle of order 3 and $S^i_{jk} = 3$, which is the smallest value of S^i_{jk}. If no paths exist that connect vertices j and k except for that one that crosses vertex i, then vertices i, j, and k form a tree structure. In this case, there is no closed loop passing through the three vertices i, j, and k, in a way that $S^i_{jk} = \infty$.

Definition 2.49. Global cyclic coefficient: The global cyclic coefficient, θ, is equal to the average of cyclic coefficients of all of the vertices, as follows [39]:

$$\theta = \frac{1}{V} \sum_{i \in \mathscr{V}} \theta_i. \tag{2.38}$$

The global cyclic coefficient takes a value between 0 and $1/3$, where 0 means the network has a tree structure in which no cycle can be found, and the opposite case ($\theta = 1/3$) indicates that there is a connection between all pairs of vertices, in which case the clustering coefficient is 1.

Definition 2.50. Modularity: The modularity measure quantifies how good a particular division of a network is [15, 57] and is designed to measure the strength of division of a network into modules (also called groups, clusters or communities). Generally, it ranges from 0 to 1. When the modularity is near 0, it means that the network does not present community structure, suggesting that the links are disposed at random in the network. As the modularity grows, the community structure gets more and more defined, that is, the mixture between communities gets smaller and therefore the fraction of links inside communities is larger than that between different communities.

Besides the network, the modularity takes as input a hypothesis about the membership of each vertex towards a community. It then tests how those vertices inside the given network fit into well-defined communities using the aforementioned notion. Mathematically, the modularity in non-weighted networks is expressed as:

$$Q = \frac{1}{2E} \sum_{i,j \in \mathscr{V}} \left(\mathbf{A}_{ij} - \frac{k_i k_j}{2E} \right) \mathbb{1}_{[c_i = c_j]}, \tag{2.39}$$

in which E represents the total number of edges in the network; k_i stands for the degree of the vertex i; c_i is the community of vertex i; and $\mathbf{A}_{ij}$ is the edge weight linking vertex i to j. The summation term is composed of two factors, all of which are computed only for vertices of the same community due to the indicator function. That is, cross-community links do not contribute to the modularity measure. The first term, $\frac{\mathbf{A}_{ij}}{2E}$, counts the fraction of links inside pairs of vertices that are members of the same community. From that, we subtract $\frac{k_i k_j}{(2E)^2}$, the second term, which accounts for removing the fraction of edges that are expected to occur due to randomness, using a random network model (recall Sect. 2.2.1). Nonzero values of the modularity index indicate deviations from randomness and values around 0.3 or more usually indicate good divisions.

We can also define the modularity for weighted networks [56]. In this case, the terms denoting the degree k_i in (2.39) are exchanged for the strength measures s_i, as introduced in Definition 2.14, and E is given by:

$$E = \frac{1}{2} \sum_{i \in \mathscr{V}} s_i. \tag{2.40}$$

The main idea of modularity is to calculate the fraction of edges that fall within the given groups minus the expected value if edges were distributed at random. For a given division of the network vertices into some modules, modularity reflects the concentration of vertices within modules compared to a random distribution of links between all vertices, regardless of modules.

Definition 2.51. Topological overlap: The topological overlap index measures to what extent two vertices are connected to roughly the same group of other vertices in the network. In essence, the topological overlap measure evaluates how similar the direct and indirect neighborhoods of two vertices are. To calculate the topological overlap of a pair of vertices, their connections to all of the other vertices in the network are compared. If these two vertices share similar direct and indirect neighborhoods, then they have a high "topological overlap." We can adjust the depth of the neighborhood which is used in the comparison. That is, we can only compare the direct neighborhood of two vertices, up to the second order neighborhood, and so on. Specifically, the m-th order topological overlap measure is constructed by (i) counting the number of m-step neighbors that a pair of vertices share and (ii) normalizing it to assume a value between 0 and 1. The resulting vertex similarity measure is a measure of agreement between the m-step neighborhoods of two input vertices. Such a measure can be applied in a number of ways, for instance, similarity search, prediction based on k-nearest neighbors, multi-dimensional scaling and module identification by clustering.

Let $\mathscr{N}_m(i)$, $m > 0$, denote the set of vertices (excluding i itself) that is reachable from i within a shortest path of length m, i.e., $\mathscr{N}_m(i) = \{j \neq i \mid d_{ij} \leq m\}$, where d_{ij} is the geodesic distance (shortest path distance) between i and j. The m-step topological overlap is given by:

$$t_{ij}^{[m]} = \begin{cases} \frac{|\mathscr{N}_m(i) \bigcap \mathscr{N}_m(j)| + \mathbf{A}_{ij}}{\min\left[|\mathscr{N}_m(i)|, |\mathscr{N}_m(j)|\right] + 1 - \mathbf{A}_{ij}}, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}, \tag{2.41}$$

in which $\mathbf{A}_{ij}$ denotes the (i, j)-th entry of the adjacency matrix of the graph. Thus, the m-step topological overlap measures the agreement of the m-step neighborhoods between two vertices. Note that, even in the case that two vertices have the same m-step neighborhoods, the topological overlap index only assumes its maximum value when they are directly connected, i.e., when $\mathbf{A}_{ij} = 1$.

2.3.4 Centrality Measures

Centrality measures quantify how central or how important vertices or edges are inside a network. The first centrality measure that comes to our mind may be the degree of a vertex. In this way, it is natural to assume that vertices with large degrees are central to the network, while vertices with small degrees are usually peripheral or terminal ones. In spite of its simplicity, degree is widely used as a centrality measure. In many real networks, vertices with large degree are often called *hubs*. Many centrality measures have been reported by the literature. Each one is defined according to a different heuristics that ultimately lead to different conclusions about the centrality of vertices or edges.

2.3.4.1 Distance-Based Centrality Measures

We divide these types of centrality measures in two groups that are classified according to the criterion used to calculate the centrality distance [41].

Definition 2.52. Minimax criterion: The first family consists of those problems that use a minimax criterion. As an example, consider the problem of determining the location for an emergency facility such as a hospital. The main objective of such an emergency facility location problem is to find a site that minimizes the maximum response time between the facility and the site of a possible emergency.

The aim of the first problem family is to determine a location that minimizes the maximum distance to any other location in the network. Suppose that a hospital is located at a vertex $u \in \mathscr{V}$. We denote the maximum distance from u to a random vertex v in the network, representing a possible incident, as the eccentricity e_u of u. Recall that the eccentricity is given by $e_u = \max_{v \in \mathscr{V}} d_{uv}$. The problem of finding an optimal location can be solved by determining the minimum over all e_u with $u \in \mathscr{V}$. Therefore, the centrality of vertex u based on the eccentricity is:

$$c_E(u) = \frac{1}{e_u} = \frac{1}{\max_{v \in \mathscr{V}} d_{uv}}. \tag{2.42}$$

Definition 2.53. Minisum criterion: The second family of location problems optimizes a minisum criterion that is used to determine the location of a service facility like a shopping mall. The aim here is to minimize the total travel time. We denote the sum of the distances from a vertex $u \in \mathscr{V}$ to any other vertex in a graph as the total distance $\sum_{v \in \mathscr{V}} d_{uv}$. The problem of finding an appropriate location can be solved by computing the set of vertices with minimum total distance as follows:

$$c_C(u) = \frac{1}{\sum_{v \in \mathscr{V}} d_{uv}}. \tag{2.43}$$

In social network analysis, a centrality index based on this concept is called closeness. The focus lies here, for example, on measuring the closeness of a person to all other people in the network. People with a small total distance are considered as more important as those with a high total distance.

2.3.4.2 Path- and Walk-Based Centrality Measures

Centrality measures that are based on paths do not take into consideration the distances from vertex to vertex, but they consider the flow passing through a vertex. In essence, a vertex is declared as more important if there are many shortest paths passing through it.

Definition 2.54. Betweenness: The betweenness measures the extent to which a vertex lies on the shortest paths between every pair of vertices in a network [29, 30, 58]. Suppose we have a network in which the vertices exchange messages among themselves. Let us initially make the simple assumption that every pair of vertices in the network exchanges a message with equal probability per unit time and that messages always take the shortest (geodesic) path of the network, or one of such paths, chosen at random, if there are several. Then, let us ask the following question: if we wait a suitably long time until many messages have passed between each pair of vertices, how many messages, on average, will have passed through each vertex *en route* to their destination? The answer is that, since messages are passing down each geodesic path at the same rate, the number passing through each vertex is simply proportional to the number of geodesic paths the vertex lies on [58]. This number of geodesic paths is what it is called betweenness index.

Given this definition, it follows that vertices with high betweenness may have considerable influence within a network by virtue of their control ability over information passing between others. The vertices with the highest betweenness in our message-passing scenario are the ones through which the largest number of messages pass, and if those vertices get to see the messages in question as they pass, or if they get paid for passing the messages along, they could derive a lot of power from their position within the network. The vertices with the highest betweenness are also the ones whose removal from the network will most disrupt communications between other vertices because they lie on the path of several messages. In real-world situations, of course, not all vertices exchange communications with the same frequency, and in most cases, communications do not always take the shortest path, due to, for example, political or physical reasons.

Mathematically, let η_{st}^{v} be 1 if vertex v lies on the geodesic path from s to t and 0 if it does not or if there is no such path (because s and t lie in different components of the network). Then, the betweenness centrality x_v is given by:

$$B_v = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{\eta_{st}^{v}}{\eta_{st}}, \tag{2.44}$$

i.e., the betweenness of v evaluates the fraction of shortest paths between all pairs of vertices s and t that passes through v over the total number of shortest paths between s and t.

Definition 2.55. Communicability [25]: Many topological and dynamical properties of complex networks are defined by assuming that most of the transport on the network flows along the shortest paths, such as the betweenness measure. However, there are different scenarios in which non-shortest paths are used to reach the network destination. For instance, in air transportation, airplanes may have to fly through more distant routes between two destinations, because in the shortest path between them there is a war or no-fly zone. Thus the consideration of only the shortest paths does not account for the global communicability of a complex network. Communicability is defined for every pair of vertices $p \in \mathscr{V}$ and $q \in \mathscr{V}$. In essence, it quantifies how easily vertex p can communicate with q by means of a combination of shortest paths and random walks with varying lengths. Mathematically, the communicability of vertex p to q is given by:

$$\mathbf{G}_{pq}(\mathbf{M}) = \frac{1}{s!}\mathbf{P}_{pq} + \sum_{k>s} \frac{1}{k!}(\mathbf{A}^{k})_{pq} = (e^{\mathbf{A}})_{pq}, \tag{2.45}$$

in which $\mathbf{P}_{pq}$ denotes the number of paths with the shortest length from p to q; s is the length of such paths; and $\mathbf{A}$ is the binary adjacency matrix of the network. The term $\mathbf{A}_{pq}^{(k)}$ is the element (p, q) of the k-th power of matrix $\mathbf{A}$ that gives the number of walks of length k from p to q along the adjacency matrix $\mathbf{A}$, with k strictly greater than s steps. The communicability of $\mathbf{G}_{pq}$ and $\mathbf{G}_{qp}$ may be different for directed graphs. A large $\mathbf{G}_{pq}$ reveals that p can reach q by several routes. Conversely, when $\mathbf{G}_{pq}$ is small, there are few possibilities for p to reach q.

2.3.4.3 Vitality

Let $\mathscr{Q}$ be the set of all simple, undirected and non-weighted graphs $\mathscr{G} = \langle \mathscr{V}, \mathscr{E} \rangle$ and let $f : \mathscr{G} \to \mathbb{R}$ be any real-valued function on $\mathscr{G} \in \mathscr{Q}$. A vitality index $V(\mathscr{G}, u)$, $u \in \mathscr{V}$, is then defined as the difference of the values of f on $\mathscr{G}$ and on $\mathscr{G}$ without element or vertex u: $V(\mathscr{G}, u) = f(\mathscr{G}) - f(\mathscr{G} \backslash \{u\})$ [41].

Definition 2.56. Flow betweenness vitality: define the max-flow betweenness vitality for a vertex $u \in \mathscr{V}$ by:

$$BV(u) = \sum_{\substack{s,t \in \mathscr{V} \\ u \neq s, u \neq t}} \frac{f_{st}(u)}{f_{st}}, \tag{2.46}$$

in which $f_{st}(u)$ is the amount of flow which must go through u. We determine $f_{st}(u)$ by $f_{st}(u) = f_{st} - \tilde{f}_{st}$ where $\tilde{f}_{st}$ is the maximal s-t-flow in $\mathscr{G} \backslash \{u\}$. That is, $\tilde{f}_{st}$ is determined by removing u from $\mathscr{G}$ and computing the maximal s-t-flow in the resulting reduced network $\mathscr{G} \backslash \{u\}$.

Definition 2.57. Closeness vitality: Let the distance between two vertices s and t represent the costs of sending a message from s to t. Then, the closeness vitality of u denotes how much the transport costs in an all-to-all communication will increase if the corresponding element u is removed from the network. That is,

$$CV(u) = I(\mathscr{G}) - I(\mathscr{G} \backslash \{u\}), \tag{2.47}$$

in which $I(\mathscr{G}) = \sum_{v,w \in \mathscr{V}} d_{vw}$, i.e, the total distance of the network.

Definition 2.58. Dynamical vitality [67]: Consider a network as a directed graph with V vertices, $\mathbf{A}u = \lambda u$ and $v^T \mathbf{A} = \lambda v^T$, where $\mathbf{A}$ is the adjacency matrix, λ is the largest eigenvalue of $\mathbf{A}$, u and v are right and left eigenvectors of $\mathbf{A}$. The dynamic importance of edge (i,j), DI_{ij}, is defined as:

$$DI_{ij} = -\frac{\Delta \lambda_{ij}}{\lambda}, \tag{2.48}$$

that is, it is the amount $-\Delta \lambda_{ij}$ by which λ decreases upon the removal of edge (i,j), normalized by λ. Similarly, the dynamical importance of vertex k is defined in terms of the amount $-\Delta \lambda_k$ by which λ decreases upon removal of that vertex:

$$DI_k = -\frac{\Delta \lambda_k}{\lambda}. \tag{2.49}$$

By removing the edge (i,j), we get $(\mathbf{A} + \Delta \mathbf{A})(u + \Delta u) = (\lambda + \Delta \lambda)(u + \Delta u)$. If we multiply by v^T, expand the formula, and neglect second order terms $v^T \Delta \mathbf{A} \Delta u$ and

$\Delta\lambda v^T \Delta u$, we obtain $\Delta\lambda = \frac{v^T \Delta \mathbf{A} u}{v^T u}$. Upon the removal of edge (i,j), the perturbation matrix is $(\Delta\mathbf{A})_{lm} = -A_{ij}\delta_{il}\delta_{jm}$, and therefore:

$$\widehat{DI}_{ij} = -\frac{\mathbf{A}_{ij} v_i u_j}{\lambda v^T u}. \tag{2.50}$$

By removing the vertex k, the perturbation matrix is given by $(\Delta\mathbf{A})_{lm} = -\mathbf{A}_{ij}(\delta_{il} + \delta_{jm})$, since $\Delta u_k = -u_k$,[2] therefore, we set $\Delta u = \delta u - u_k e_k$, where e_k is the unit vector for the k-th component, and we assume that δu is small. By multiplying v^T and again neglecting the second order terms $v^T \Delta \mathbf{A} \delta u$ and $\Delta\lambda v^T \delta u$, we obtain $\Delta\lambda = \frac{(v^T \Delta \mathbf{A} u - u_k v^T \Delta \mathbf{A} e_k)}{(v^T u - v_k u_k)}$. Using the expression of $\Delta\mathbf{A}$, we get $v^T \Delta \mathbf{A} u = -2\Delta u_k v^k$ and $u_k v^T \Delta \mathbf{A} e_k = \lambda u_k v_k$. Considering that the network is large ($V \gg 1$), we assume that $u_k v_k < v^T u$. Thus, we obtain:

$$\widehat{DI}_k = -\frac{v_k u_k}{\lambda v^T u}. \tag{2.51}$$

2.3.4.4 General Feedback Centrality

Now we present measures that are built on the concept of feedback centrality. In this respect, a vertex has larger feedback centrality the more central are its neighbors [41].

Definition 2.59. Bonacich's eigenvector centrality: In 1972, Phillip Bonacich [9] introduced a centrality measure that is computed using eigenvectors of adjacency matrices. In special, he presented three different approaches to evaluate the centrality measure and all three of them result in the same valuation of the vertices. The difference between these methodologies are in a constant factor. In the following, we assume that the graph $\mathcal{G}$ is undirected, connected, without self-loops, and non-weighted. As the graph is undirected and without self-loops, the adjacency matrix $\mathbf{A}$ is symmetric and all diagonal entries are zero. The three methods that score each vertex are:

1. The factor analysis approach;
2. The convergence of an infinite sequence; and
3. The solution of a simultaneous linear equation system.

Here, we only focus on the third approach. It follows the idea of calculating an eigenvector of a linear equation system. If we define the centrality of a vertex to be a weighted sum of the centralities of its adjacent vertices, where the weight is given by the network topology, we get the following equation system:

[2] Recall that the left and right eigenvectors have zero k-th entries after the removal of vertex k.

$$s_i = \sum_{j \in \mathscr{V}} \mathbf{A}_{ij} s_j, \tag{2.52}$$

in which s_i is the Bonacich score or centrality of vertex i. In a matrix form,

$$s = \mathbf{A}s. \tag{2.53}$$

Equation (2.53) has a single solution only if $\det(\mathbf{A} - \mathbf{I}) = 0$, where $\mathbf{I}$ is the identity matrix. We can instead solve for s using the eigenvalue problem of $\mathbf{A}$, i.e., $\lambda s = \mathbf{A}s$.

Definition 2.60. Katz index: This index first appeared in the context of social networks to determine the importance or status of an individual [38]. To take the number of intermediate individuals into account, a damping factor $\alpha > 0$ is introduced: the longer the path between two vertices i and j is, the smaller should its impact on the status of j be. The associated mathematical model is hence a non-weighted, directed graph $\mathscr{G} = \langle \mathscr{V}, \mathscr{E} \rangle$ without self-loops and associated adjacency matrix $\mathbf{A}$. Using the fact that $(\mathbf{A}^k)_{ji}$ holds the number of paths from j to i with length k, the status of vertex i is:

$$C_k(i) = \sum_{k=1}^{\infty} \sum_{j \in \mathscr{V}} \alpha^k (\mathbf{A}^k)_{ji}. \tag{2.54}$$

In matrix notation, we have:

$$C_K = \sum_{k=1}^{\infty} \alpha^k (\mathbf{A}^T)^k \mathbf{1}_V, \tag{2.55}$$

in which $\mathbf{1}_V$ is the V-dimensional vector where every entry is 1. Assuming that $\alpha|\lambda_0| < 1$, where λ_0 is the largest eigenvalue of $\mathbf{A}$, the infinite series converges. Thus, we can find a closed form expression for the status index of Katz:

$$C_K = \sum_{k=1}^{\infty} \alpha^k (\mathbf{A}^T)^k \mathbf{1}_V = (\mathbf{I} - \alpha \mathbf{A}^T)^{-1} \mathbf{1}_V \tag{2.56}$$

or in another form:

$$(\mathbf{I} - \alpha \mathbf{A}^T) C_K = \mathbf{1}_V, \tag{2.57}$$

which is an inhomogeneous system of linear equations that emphasizes the feedback nature of the centrality: the value of $C_K(i)$ depends on the centrality values of neighbors of i in the graph, i.e., $C_K(j), j \neq i$.

Definition 2.61. Web page centrality—PageRank: PageRank (PR) is a well-known measure used by Google to rank web pages. It is supposed to simulate the behavior of a user browsing the Web. Most of the time, the user visits pages just by surfing, i.e., by clicking on hyperlinks of the page he/she is on. Another manner is to jump to another page by typing its URL on the browser, or going to a bookmark, etc. In a network, this process can be modeled by a simple combination of a random walk with occasional jumps toward randomly selected vertices. This can be described by the simple set of implicit relations [64]:

$$p(i) = \frac{q}{V} + (1-q) \sum_{j \in \mathscr{V} : j \to i} \frac{p(j)}{k_j^{(\text{out})}}. \tag{2.58}$$

Here, V is the number of vertices of the graph, $p(i)$ is the PR value of vertex i, $k_j^{(\text{out})}$ the out-degree of vertex j, and the sum runs over the vertices pointing toward (direct connection to) i. The damping factor $q \in [0,1]$ is a probability that weighs the mixture between the realized random walk and random jumps.

For any $q > 0$, the process reaches stationarity, as a walker has a finite (no matter how small) probability to escape from a dangling end, whenever it lands there. When $q = 0$, the process may not be stationary and PR is ill defined. When $q = 1$, instead, the jumping process dominates and all of the vertices have the same PR-value $1/V$.

PR goes beyond the concept of in-degree. In order to have a large PR for a vertex, it is important to have many neighbors pointing at that vertex, i.e., large in-degree, but it is also important that the neighbors have large PR values themselves. So, if two vertices have equal in-degree, the vertex with more "important" neighbors will have larger PR.

Definition 2.62. Eigenvector centrality: The eigenvector centrality, like the PageRank, relies on the principle that the importance of a vertex depends on the importance of its neighbors [64]. The relationship that the eigenvector centrality captures is more straightforward than that in PageRank: the prestige x_i of vertex i is simply proportional to the sum of the prestiges of the neighboring vertices pointing to it. Numerically,

$$\lambda x_i = \sum_{j \in \mathscr{V} : j \to i} x_i = \sum_{j \in \mathscr{V}} \mathbf{A}_{ji} x_j = (\mathbf{A}^T x)_i. \tag{2.59}$$

We see that x_i is basically the i-th component of the transposed eigenvector of the adjacency matrix $\mathbf{A}$ associated to the eigenvalue λ. We observe that the trivial eigenvector with all of the components equal to zero is always a solution of (2.59). From (2.59), we also see that singleton vertices have zero centrality. In general, vertices pointed at by vertices with zero centrality also have zero centrality and this effect will propagate to other vertices, so that in many cases the eigenvector centrality would not give any information about a large fraction of vertices. To avoid this, it is useful to make the following modification: to each vertex, we assign a prestige ϵ, which is independent of its relationships with the other vertices. As a result, Eq. (2.59) becomes [64]:

$$x_i = \alpha(\mathbf{A}^T x)_i + \epsilon. \tag{2.60}$$

The role of the parameter ϵ reminds that of the damping factor q in PageRank. The parameter α weighs the relative importance of the contribution of the peers versus that of the vertex itself.

2.3.5 Classification of the Network Measurements

As it can be noticed, the complex network literature has proposed a myriad of network measurements that capture different aspects of the network structure. The provided list is far from being exhaustive. New network measurements are introduced to suit the needs of computational problems that arise in our day-to-day problems. Some of them may be domain-dependent and others may even require external information to be computed. In the previous sections, we have introduced the network measurements by dividing them into functional roles. In this section, we re-compile these network measurements using a meta-information approach. We classify them in accordance with the type of information they use in their computation. We define three classes of network measurements, as follows:

- *Strictly local measures*: these measures only employ information from the vertex itself to be computed. Strictly local measures are always vertex-level measures.
- *Mixed measures*: besides using strictly local information, these measures also use topological information from its direct and indirect neighborhoods. This additional information can vary from simply quasi-local topology, such as the number of triangles in the neighborhood, to long-range information, such as the shortest path between the two most distant pair of vertices. Mixed measures are always vertex-level measures.
- *Global measures*: these network measurements make use of the entire network structure to be computed. Global measures are always network-level measures.

Figure 2.21 portrays a schematic of the three classes of network measurements. Strictly local and mixed measures are vertex-level, while global measures must be network-level. Table 2.1 reports the classification of the network measurements we have discussed so far in this chapter.

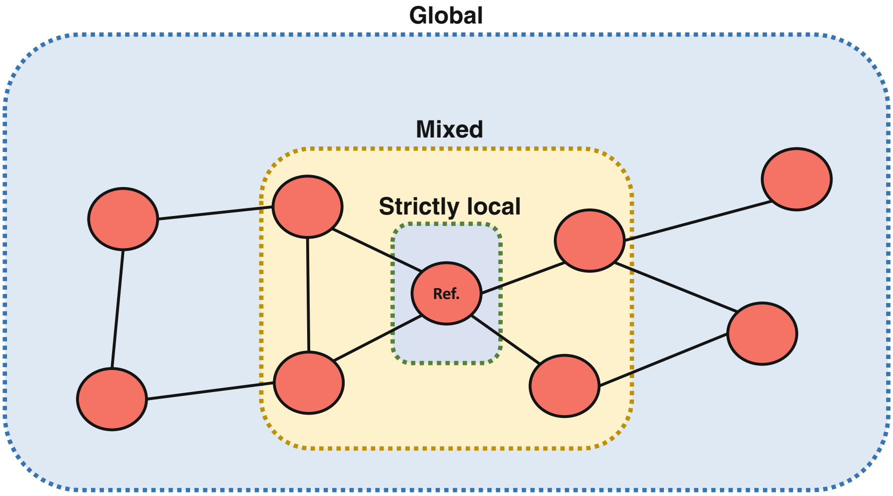

Fig. 2.21 Intuition for classifying network measurements in terms of the type of information they need to be computed

2.4 Dynamical Processes in Complex Networks

One of the fundamental differences between graph theory and complex network studies is that the latter focus not only on the static structures but also on the dynamical properties of networks under study. Therefore, in this section, we review five dynamical processes in networks: random walk, lazy random walk, self-avoiding walks, tourist walk, and epidemic spreading. Besides these ones, there are many other dynamical processes in networks, such as information transmission, percolation in regular lattices and in complex networks, and synchronization among oscillators (vertices). However, the last ones are not the focus of this book.

2.4.1 Random Walks

A random walk is a mathematical formalization of a trajectory that consists of taking successive random steps [63]. It has been used to describe many natural phenomena and it has also been applied to solve a wide range of engineering problems. Some of these include graph matching and pattern recognition [33], image segmentation [34], neural network modeling [37, 47], network centrality measure [59], network partition [81], construction and analysis of communication networks [78, 80].

Given a network $\mathscr{G} = \langle \mathscr{V}, \mathscr{E} \rangle$ and a starting vertex $v \in \mathscr{V}$, we select a neighbor of it at random, and move to this neighbor; then we select a neighbor of this new

Table 2.1 Classification of the network measurements using a meta-information approach

Definition	Description	Classification
2.10	Degree	Strictly local
2.11	In- and out-degree	Strictly local
2.12	Average degree (connectivity)	Global
2.13	Average in- and out-degree	Global
2.14	Strength	Strictly local
2.15	In- and out-strength	Strictly local
2.35	Density	Global
2.36	Assortativity	Global
2.37	Local assortativity	Mixed
2.38	Non-normalized rich-club coefficient	Global
2.39	Normalized rich-club coefficient	Global
2.40	Diameter	Global
2.41	Vertex eccentricity	Mixed
2.42	Radius	Global
2.43	Wiener index	Global
2.44	Global efficiency	Global
2.45	Average harmony	Global
2.46	Clustering coefficient	Mixed
2.47	Network clustering coefficient	Global
2.48	Cyclic coefficient	Mixed
2.49	Global cyclic coefficient	Global
2.50	Modularity	Global
2.51	Topological overlap	Mixed
2.52	Eccentricity centrality (minimax criterion)	Mixed
2.53	Total distance centrality (minisum criterion)	Mixed
2.54	Betweenness	Mixed
2.55	Communicability	Mixed
2.56	Flow betweenness vitality	Mixed
2.57	Closeness vitality	Mixed
2.58	Dynamical vitality	Mixed
2.59	Bonacich centrality	Mixed
2.60	Katz index	Mixed
2.61	PageRank	Mixed
2.62	Eigenvector centrality	Mixed

vertex again at random, and move to it, and so on. The random sequence of vertices selected this way is a random walk on the graph. A finite random walk of length $t > 0$ has the same intuition, but we stop after making $t - 1$ random transitions. If the graph is weighted, then we transition to a neighbor u with probability that is proportional to the edge weight $\mathbf{A}_{vu}$.

In essence, the theory of random walks on networks and the theory of finite discrete Markov chains are basically the same, so that every discrete Markov chain can be conceived as random walk on a graph. Discrete Markov chains are stochastic processes whose future states are conditionally independent of the past states provided that the present state is known. In graph theory, the states are denoted by the vertices in the graph. In a graph theory context, given that a walker is at vertex v, the Markovian property affirms that the probability of visiting a neighboring vertex is independent on the past trajectories of that walker. We formalize that concept in the following.

Definition 2.63. Discrete-time Markov chain: A discrete-time Markov chain is a stochastic process $\{X_t : t \in \mathbb{N}\}$, where the random variable X assumes values in a countable set $\mathcal{N}$ at any given time t. The transition probability to state $q \in \mathcal{N}$ is:

$$P[X_t = q \mid X_{t-1}, X_{t-2}, \ldots, X_0] = P[X_t = q \mid X_{t-1}], \tag{2.61}$$

i.e., the probability of the next outcome only depends on the last value of the process. Therefore, past trajectories are irrelevant.

Remark 2.32. In the context of graph theory, the countable set is composed of the vertex set, i.e., $\mathcal{N} = \mathcal{V}$.

Remark 2.33. In Markovian processes, each feasible value in the countable set $\mathcal{V}$ is called a state.

Definition 2.64. Transition probability: The transition probability of going from state (vertex) q to u is denoted by $\mathbf{P}_{qu}(t)$, $q, u \in \mathcal{V}$, which is a shorthand for $\mathbf{P}_{qu}(t) = P[X_t = u \mid X_{t-1} = q]$. Mathematically, the transition probability is defined in accordance with the network topology, as follows:

$$\mathbf{P}_{qu} = \frac{\mathbf{A}_{qu}}{\sum_{i \in \mathcal{V}} \mathbf{A}_{qi}}, \tag{2.62}$$

i.e., the stronger is the edge weight linking q to u, the more likely will be that transition.

Remark 2.34. Rewrite (2.62) as:

$$\mathbf{P}_{qu} = \frac{\mathbf{A}_{qu}}{K(q)}, \tag{2.63}$$

in which $K(q) = \sum_{i \in \mathscr{V}} \mathbf{A}_{qi}$. Then,

- If the network is undirected and non-weighted, then $K(q) = k_q$, where k_q is the degree of vertex q.
- If the network is directed and non-weighted, then $K(q) = k_q^{(\text{out})}$, where $k_q^{(\text{out})}$ is the out-degree of vertex q.
- If the network is undirected and weighted, then $K(q) = s_q$, where s_q is the strength of vertex q.
- If the network is directed and weighted, then $K(q) = s_q^{(\text{out})}$, where $s_q^{(\text{out})}$ is the out-strength of vertex q.

Definition 2.65. Transition matrix: In Markovian processes, we can map all of the feasible transitions using the transition matrix $\mathbf{P}(t)$ as follows:

$$\mathbf{P}(t) = \begin{pmatrix} \mathbf{P}_{1,1}(t) & \mathbf{P}_{1,2}(t) & \dots & \mathbf{P}_{1,V}(t) \\ \mathbf{P}_{2,1}(t) & \mathbf{P}_{2,2}(t) & \dots & \mathbf{P}_{2,V}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{V,1}(t) & \mathbf{P}_{V,2}(t) & \dots & \mathbf{P}_{V,V}(t) \end{pmatrix}. \tag{2.64}$$

Note that the transition matrix completely characterizes the Markovian process because, the immediate future state $X(t+1)$ is only determined by the current state $X(t)$, regardless of the past trajectories.

Remark 2.35. If $\mathbf{P}(t)$ is immutable for all $t \in \mathbb{N}$, then the Markov process is said to be *time-homogenous*. In a graph theory perspective, this is equivalent to saying that the graph topology does not change during the walk. For clarity, if the Markov process (or random walk) is time-homogeneous, we drop the time indexing of the transition matrix.

Definition 2.66. *m*-step transition matrix: For a time-homogeneous Markovian process, we can define the m-step transition matrix, $m > 0$, as $\mathbf{P}^m$. Essentially, the entry $\mathbf{P}^m_{qu}$ encodes the transition probability of starting from state or vertex q and arriving at state or vertex u after exactly m transitions.

Remark 2.36. The original transition matrix defined in (2.64) is a 1-step transition matrix.

For each realization of the Markovian process $\omega \in \Omega$, let $\text{pt}(j)$ be the number of times j appears in the random walk that visits the states $X_0(\omega), X_1(\omega), X_2(\omega), \dots$. Then, $\text{pt}(j)$ is the total number of times the state j is visited by the stochastic process X in realization ω. If $\text{pt}(j)$ is finite, then X eventually leaves state j never to return. Mathematically, there must be an integer n such that $X_n(\omega) = j$ and $X_m(\omega) \neq j$, $\forall m > n$. In contrast, if $\text{pt}(j) = \infty$ for a realization ω, then X keeps on visiting j again and again. These two classes that state j can assume are important from a practical point-of-view [14]. We now turn our attention in providing formal tools to classify states according to those perspectives.

The passage time function counts the number of times a given vertex has been visited during a random walk. We formalize this notion in the following.

Definition 2.67. Passage Time: The passage time is a function pt : $\mathscr{V} \to \mathbb{N}$ such that pt(q) is the number of times the Markovian process reaches the state q. Mathematically,

$$\begin{aligned} \mathrm{pt}(q) &= |\{t \in \mathbb{N} \mid X_t = q\}| \\ &= \sum_{t=0}^{\infty} \mathbb{1}_{[X_t(\omega)=q]}. \end{aligned} \tag{2.65}$$

Recall that $\mathbb{1}_{[A]}$ is the indicator function that yields 1 whenever the logical expression A is true, and returns 0, otherwise. Basically, we increment pt(q) by one each time the stochastic process X visits state or vertex q.

We now define the so-called potential matrix of the Markovian process X.

Definition 2.68. Potential or fundamental matrix: The potential matrix $\mathbf{R}$ encodes the expected number of times each vertex is visited when we start from any given other vertex. Mathematically, its (i, j)-th entry is expressed as:

$$\mathbf{R}_{ij} = \mathbb{E}\left[\mathrm{pt}(j) \mid X(0) = i\right], \tag{2.66}$$

which can be seem as the mean passage time to reach j conditioned that the walker starts at vertex i.

Plugging (2.65) into (2.66) and using the monotone convergence theorem, we get:

$$\begin{aligned} \mathbf{R}_{ij} &= \mathbb{E}\left[\sum_{n=0}^{\infty} \mathbb{1}_{[X_n=j]} \,\middle|\, X(0) = i\right] \\ &= \sum_{n=0}^{\infty} \mathbb{E}\left[\mathbb{1}_{[X_n=j]} \,\middle|\, X(0) = i\right] \\ &= \sum_{n=0}^{\infty} P\left(X_n = j \,\middle|\, X(0) = i\right) \\ &= \sum_{n=0}^{\infty} \mathbf{P}_{ij}^{m}. \end{aligned} \tag{2.67}$$

Let T be the time that state or vertex j is first visited by a realization of the Markovian process.

Definition 2.69. Recurrent state: State j is recurrent if:

$$P(T < \infty \mid X(0) = j) = 1. \tag{2.68}$$

As a consequence, the number of returns of a recurrent state is always infinite, that is:

$$\mathbf{R}_{jj} = \mathbb{E}\left[\mathrm{pt}(j) \mid X(0) = j\right] = \infty. \tag{2.69}$$

Definition 2.70. Transient state: State j is transient if:

$$P(T = +\infty \mid X(0) = j) > 0. \tag{2.70}$$

As a consequence, the number of returns of a transient state is always finite, that is:

$$\mathbf{R}_{jj} = \mathbb{E}\left[\mathrm{pt}(j) \mid X(0) = j\right] < \infty. \tag{2.71}$$

Remark 2.37. There are only two states: recurrent or transient. In this way, if j is not recurrent, then it must be a transient state, and vice versa.

Remark 2.38. Let j be a recurrent state. Then, we sub-classify it as *null recurrent* if:

$$\mathbb{E}[T \mid X(0) = j] = \infty, \tag{2.72}$$

otherwise, we call it *non-null recurrent.*

Remark 2.39. Let j be a recurrent state. Then, we sub-classify it as *periodic with period δ* if $\delta \geq 2$ is the largest integer for which:

$$P(T = n\delta \text{ for some } n \geq 1) = 1. \tag{2.73}$$

otherwise, we call it *aperiodic*.

Definition 2.71. Closed set of states: A set of states is said to be closed if no state outside it can be reached from any state inside it.

Definition 2.72. Absorbing state: A state forming a closed set by itself is called an absorbing state. We say that state q is absorbing if there is a probability 1 to go from q to itself. In other words, once an absorbing state has been reached in a random walk, the walker stays in this state forever.

Definition 2.73. Irreducible closed set: A closed set is irreducible if no proper subset of it is closed.

Definition 2.74. Irreducible Markov chain: A Markov chain is called irreducible if its only closed set is the set of all states. Therefore, a Markov chain is irreducible if and only if all states can be reached from each other.

The state set of the Markov chain process can be divided into the absorbing state set $\mathscr{V}_A$ and its complementary set, the transient state set $\mathscr{V}_T = \mathscr{V} \backslash \mathscr{V}_A$.

Remark 2.40. The mean passage time for transient states can be obtained by computing the fundamental matrix only for the transient states $\mathbf{R}^{(\text{transient})}$:

$$\mathbf{R}^{(\text{transient})} = (\mathbf{I} - \mathbf{P}_T)^{-1}, \tag{2.74}$$

in which $\mathbf{I}$ is the $|\mathscr{V}_T| \times |\mathscr{V}_T|$ identity matrix and $\mathbf{P}_T$ is the transition probability matrix restricted to the transient states. The entry $\mathbf{R}^{(\text{transient})}_{q'q}$ contains the mean passage time in state $q \in \mathscr{V}_T$ during random walks starting in state q'. Hence,

$$\mathbb{E}\left[\text{pt}(q)\right] = [p'^{(\text{transient})}\mathbf{R}^{(\text{transient})}]_q, \tag{2.75}$$

in which $p'^{(\text{transient})}$ is the transpose of the initial probability vector when we only consider transient states. Note that the expectation operation is taken over random walks with arbitrary lengths.

Given a distribution $p(t)$, $\dim(p(t)) = 1 \times V$, where the v-th entry denotes the probability that the system will be at vertex $v \in \mathscr{V}$, the evolution of $p_v(t)$ is:

$$p_v(t+1) = \sum_{(u,v) \in \mathscr{E}} \mathbf{P}(t)_{uv} p_u(t). \tag{2.76}$$

Analogously, the evolution of the distribution $p(t)$ is:

$$p(t+1) = p(t)\mathbf{P}(t). \tag{2.77}$$

Intuitively, the evolution of the probability distribution $p(t)$ as a function of t can be seen as describing a diffusion process in the underlying graph. The diffusion is completely characterized once we know the initial distribution $p(0)$ and the transition matrices $\mathbf{P}(t)$.

Definition 2.75. Stationary distribution: If the network $\mathscr{G}$ is a finite, irreducible, time-homogenous, and aperiodic Markov chain, then it has a unique stationary distribution $\pi = [\pi_1, \ldots, \pi_V]$ that can be reached from any initial distribution $p(0)$. In the dynamic equation, the stationarity is reached when the following holds:

$$\pi = \pi\mathbf{P}. \tag{2.78}$$

Each entry of the stationary distribution is of the form:

$$\pi_i = \frac{1}{\mathbb{E}\left[T \mid X(0) = i\right]}, \tag{2.79}$$

in which recall that $\mathbb{E}\left[T \mid X(0) = i\right]$ is the expected time to regress to vertex i starting from i.

For an undirected network, we have that:

$$\mathbb{E}\left[T \mid X(0) = i\right] = \frac{\sum_{j \in \mathscr{V}} k_j}{k_i} == \frac{2E}{k_i}, \tag{2.80}$$

in which E is the number of edges in the network and k_i is the degree of vertex i.

Substituting (2.80) in (2.79), we get:

$$\pi_i = \frac{k_i}{2E}. \tag{2.81}$$

2.4.2 Lazy Random Walks

The unique stationary distribution in Definition 2.75 only holds true, among other things, for aperiodic networks. However, if the network is periodic, there is an easy way to fix the periodicity problem by introducing the lazy random walk. In a lazy random walk at time t, the walker may decide upon two different actions:

1. It can transition to a neighboring vertex in accordance with the transition matrix with probability $1/2$; or
2. It can stay at the current vertex[3] with probability $1/2$.

Remark 2.41. The lazy random walk can be viewed as a vanilla version of the classical random walk in a network in which we add k_u self-loops to every vertex u in the original graph $\mathscr{G}$.

Formally, the evolution of the probability distribution $p(t)$ of a lazy random walk is given by:

$$\begin{aligned} p(t+1) &= \frac{1}{2}p(t) + \frac{1}{2}p(t)\mathbf{P}(t) \\ &= p(t)\frac{1}{2}\left[\mathbf{I} + \mathbf{P}(t)\right] \\ &= p(t)\mathbf{P}'(t), \end{aligned} \tag{2.82}$$

[3] Hence, the terminology "lazy" random walk.

in which $\mathbf{P}'(t)$ is the modified transition matrix for the lazy random walk:

$$\mathbf{P}'(t) = \frac{1}{2}\left[\mathbf{I} + \mathbf{P}(t)\right]. \tag{2.83}$$

Note also that the stationary distribution of a lazy random walk is identical to that of the classical random walks portrayed in Definition 2.75. To see that, it suffices to see that $\mathbf{P}'(t)$ is also a valid transition matrix, just like the original $\mathbf{P}(t)$. As long as the graph $\mathscr{G}$ is finite, irreducible, time-homogenous, and aperiodic, the unique stationary distribution always exists.

2.4.3 Self-Avoiding Walks

A self-avoiding walk on a network $\mathscr{G}$ is a path that visits no vertex more than once. Self-avoiding walks were first introduced in the chemical theory of polymerization [27], and since then their critical behavior has attracted attention of mathematicians and physicists [49].

Broadly speaking, self-avoiding walks are usually considered in infinite lattices, so that steps are only allowed in a discrete number of directions and of certain lengths. Self-avoiding walks cannot be Markovian, because we need to check the past trajectory in order to list the possible futures states that the process can assume. The research in [49] provides a comprehensive review on self-avoiding walks.

2.4.4 Tourist Walks

A tourist walk can be conceptualized as a walker (tourist) aiming at visiting sites (data items) in a P-dimensional map, representing the data set. At each discrete timestep, the tourist follows a simple deterministic rule: it visits the nearest site that has not been visited in the previous μ steps. In other words, the walker performs partially self-avoiding deterministic walks over the data set, where this self-avoiding factor is limited to the memory window $\mu - 1$. This quantity can be understood as a repulsive force emanating from the sites in this memory window, which prevents the walker from visiting them in this interval (refractory time). Therefore, it is prohibited that a trajectory intersects itself inside this memory window. In spite of being a simple rule, it has been shown that this kind of movement possesses complex behavior when $\mu > 1$ [48]. Note that tourist walks differ from self-avoiding random walks in that the former is a deterministic process, while the latter is a random process.

The tourist's behavior heavily depends on the data set's configuration and the starting site. In computational terms, the tourist's movements are entirely realized

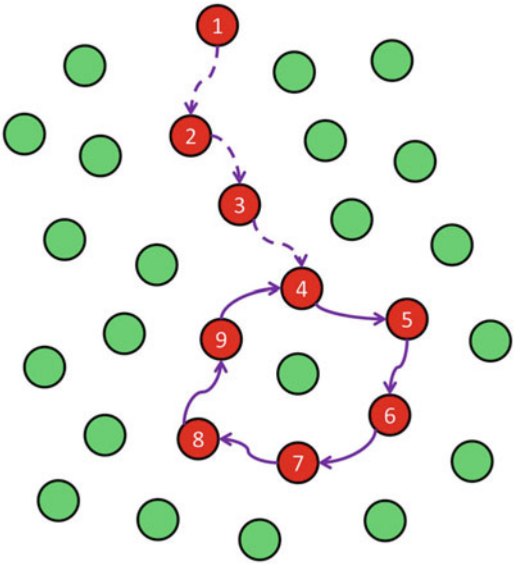

Fig. 2.22 Illustration of a tourist walk with $\mu = 1$. The *red* (*dark gray*) and *green* (*light gray*) *dots* represent visited and unvisited sites, respectively. The *dashed lines* indicate the transient part of the walk, whereas the *continuous lines,* the attractor of the walk

by means of a neighborhood table. This table is constructed by ordering all the data items in relation to a specific site. This procedure is performed for every site of the data set.

Each tourist walk can be decomposed in two terms: (1) the initial *transient part* of length t and (2) a *cycle* (attractor) with period c. Figure 2.22 shows an illustration of a tourist walk with $\mu = 1$. In this case, one can see that the transient length is $t = 3$ and the cycle length, $c = 6$.

Considering the attractor or cycle period as a walk section that begins and ends at the same site of the data set may lead one to think that, once the tourist visits a specific site, a new visit to it would configure an attractor. Nevertheless, this is a very simple, and likely to fail, approach for attractors' detection. In fact, during a walk, a site may be re-visited without configuring an attractor. Besides, the tourist's finite memory μ allows some steps of the walk to be repeated without configuring an attractor. For instance, if we had chosen a $\mu = 6$ for the walk in Fig. 2.22, the re-visit performed by the tourist on the site 4 would have not configured an attractor, since the site 5 would still be forbidden to be visited again; hence, the tourist would be compelled to visit another site. This characteristic enables sophisticated trajectories over the data set, at cost of also increasing the difficulty in detecting an attractor.

In the majority of the works related to tourist walk [40, 48, 71], the tourist may visit any other site other than the ones contained in its memory window. As μ increases, there is a significant chance that the walker will begin performing large jumps in the data set, since the neighborhood is most likely to be already visited in its entirety within the time frame μ. In the context of data classification, this is an undesirable characteristic that can be simply avoided by using a graph representation of the input data. In this way, the walker is only permitted to visit vertices, represented now by the sites, that are in its connected neighborhood (link). With this modified mechanism, for large values of μ, it is very likely that the walker will get trapped within a vertex, not being able to further visit other vertices of the neighborhood. In this scenario, we say that the walk only had a transient part and the cycle period is null ($c = 0$).

2.4.5 Epidemic Spreading

Epidemic spreading in complex networks has triggered much attention to many researchers. It is a dynamic process within a network and the main concern is how the network structure attenuates or amplifies disease breakouts or immunization. Since epidemic spreading processes can be considered as information transmission, it is useful for machine learning. For example, epidemic spreading may be directly related to data label propagation in semi-supervised learning. Although we have not found works connecting epidemic spreading and machine learning in literature yet, we would like to share such a prediction with the readers. The readers who are interested in this topic are invited to develop their new techniques in this direction. For the above-mentioned purpose, we here review two basic models of epidemic spreading in complex networks. For a comprehensive review, see [23, 62, 83]. For some development related to information transmission in complex networks, see [18, 50, 76, 77, 79].

The most extensively studies of epidemic models are about the susceptible-infected-recovered and susceptible-infected-susceptible models [3, 4, 36]. We review these models in the following.

2.4.5.1 Susceptible-Infected-Recovered (SIR) Model

In the SIR model, each individual is at one of the three states: susceptible (does not infect others but may be infected), infected, or recovered (will not be affected again). At each time step, assume that a susceptible individual may be infected by another infected person with probability α and that the recovering rate of infected individuals is β. Then, the epidemic process in the SIR model can be described by the following dynamic equations:

$$\frac{dx}{dt} = -\alpha yx, \tag{2.84}$$

$$\frac{dy}{dt} = -\alpha yx - \beta y, \tag{2.85}$$

$$\frac{dz}{dt} = -\beta y, \tag{2.86}$$

in which x, y and z are the ratios of susceptible, infected, and removed individuals to the entire population, respectively. In a network setting, each individual is represented by a vertex and links exist when two individuals have some kind of contact. In this network, a susceptible vertex will be infected only if it has at least one infected neighbor.

2.4.5.2 Susceptible-Infected-Susceptible (SIS) Model

For some diseases, such as influenza and pulmonary tuberculosis, the recovered individual can be infected again. This situation is not considered by the SIR model. For this reason, the SIS model was introduced. The only difference between them is that in SIS model, the infected individuals will return to the susceptible state after recovering. The SIS model is defined by the following equations:

$$\frac{dx}{dt} = -\alpha yx + \beta y, \tag{2.87}$$

$$\frac{dy}{dt} = \alpha yx - \beta y. \tag{2.88}$$

2.4.5.3 Epidemic Spreading in Complex Networks

In [43], the authors studied the SIS model on small-world networks of Watts and Strogatz, which have been presented in Sect. 2.2.2. They found that even when the rewiring probability p is very small (for instance, $p = 0.01$), the disease can permanently exist with very small infection ratios and without fluctuations in the population ratios. In contrast, when p gets large enough (for example, $p = 0.9$), periodic oscillations of the number of infected individuals start to appear.

Consider the SIS model in random networks and assume that λ denotes the spreading rate. In [6, 7], the authors uncovered a spreading threshold λ_c. If the value of λ is above the threshold, i.e., $\lambda > \lambda_c$, the infection spreads and becomes persistent. Below it, the infection disappears. Such a result implies that the disease can persist only if it infects a sufficiently large amount of individuals. However, in real situations, many diseases can persistently exist with just a small fraction of the population being infected, such as computer viruses and measles. In [6, 7], the authors obtained the epidemic threshold of the SIS dynamics in general networks as follows:

$$\lambda_c = \frac{\langle k \rangle}{\langle k^2 \rangle}, \tag{2.89}$$

in which $\langle . \rangle$ represents an averaging operator over all of the network vertices, and k denotes the degree. Note that $\langle k \rangle = \bar{k}$, which is the network connectivity. In scale-free networks, when the network size goes to infinite, we have that $\lambda_c = 0$. The absence of epidemic threshold in scale-free networks provides a good explanation for the empirical data [6, 7].

2.5 Chapter Remarks

In this chapter, we have introduced the basic notion of graphs and some of the network topologies that are well-known by the complex network community. We have also explored a comprehensive list of network measurements, which are able to extract structural information of the data relationships in a systematic manner. Finally, we have reviewed classical dynamic processes, such as the random walk, self-avoiding walk, tourist walk, and epidemic spreading with a focus on networked environments.

References

1. Ahn, Y.Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature **466**, 761–764 (2010)
2. Albert, R., Jeong, H., Barabási, A.L.: Diameter of the world wide web. Nature **401**, 130–131 (1999)
3. Anderson, R.M., May, R.M.: Infectious Diseases of Humans: Dynamics and Control. Oxford University Press, Oxford, NY (1992)
4. Bailey, N.: The Mathematical Theory of Infectious Diseases and Its Applications. Griffin, London (1975)
5. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science - New York **286**(5439), 509–512 (1999)
6. Boguñá, M., Pastor-Satorras, R.: Epidemic spreading in correlated complex networks. Phys. Rev. E **66**, 047104 (2002)
7. Boguñá, M., Pastor-Satorras, R., Vespignani, A.: Absence of epidemic threshold in scale-free networks with connectivity correlations. Phys. Rev. Lett. **90**, 028701 (2003)
8. Bollobas, B.: Modern Graph Theory. Springer, Berlin (1998)
9. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. J. Math. Sociol. **2**(1), 113–120 (1972)
10. Borgatti, S.P., Everett, M.G.: Models of core/periphery structures. Soc. Netw. **21**(4), 375–395 (2000)
11. Callaway, D.S., Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Network robustness and fragility: Percolation on random graphs. Phys. Rev. Lett. **85**, 5468–5471 (2000)
12. Chase-Dunn, C.K.: Global Formation: Structures of the World-Economy. Blackwell, Oxford (1989)
13. Chung, F.R.K.: Spectral Graph Theory. CBMS Regional Conference Series in Mathematics, vol. 92. American Mathematical Society, Philadelphia (1997)
14. Çinlar, E.: Introduction to Stochastic Processes. Prentice-Hall, Englewood Cliffs, NJ (1975)
15. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 066111+ (2004)
16. Cohen, R., Erez, K., ben Avraham, D., Havlin, S.: Resilience of the internet to random breakdowns. Phys. Rev. Lett. **85**, 4626–4628 (2000)
17. Cohen, R., Erez, K., ben Avraham, D., Havlin, S.: Breakdown of the internet under intentional attack. Phys. Rev. Lett. **86**, 3682–3685 (2001)
18. Cohen, R., Havlin, S., ben Avraham, D.: Efficient immunization strategies for computer networks and populations. Phys. Rev. Lett. **91**, 247901 (2003)
19. Colizza, V., Flammini, A., Serrano, M.A., Vespignani, A.: Detecting rich-club ordering in complex networks. Nat. Phys. **2**(2), 110–115 (2006)

20. Costa, L.F., Rodrigues, F.A., Travieso, G., Villas Boas, P.R.: Characterization of complex networks: a survey of measurements. Adv. Phys. **56**(1), 167–242 (2007)
21. Diestel, R.: Graph Theory. Graduate Texts in Mathematics. Springer, Berlin (2006)
22. Doreian, P.: Structural equivalence in a psychology journal network. J. Am. Soc. Inf. Sci. **36**(6), 411–417 (1985)
23. Draief, M., Massouli, L.: Epidemics and Rumours in Complex Networks. Cambridge University Press, New York, NY (2010)
24. Erdös, P., Rényi, A.: On random graphs I. Publ. Math. (Debrecen) **6**, 290–297 (1959)
25. Estrada, E., Hatano, N.: Communicability in complex networks. Phys. Rev. E **77**, 036111 (2008)
26. Everett, M., Borgatti, S.: Regular equivalence: general theory. J. Math. Sociol. **18**(1), 29–52 (1994)
27. Flory, P.: Principles of Polymer Chemistry. Cornell University Press, Ithaca (1953)
28. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
29. Freeman, L.C.: A set of measures of centrality based upon betweenness. Sociometry **40**, 35–41 (1977)
30. Freeman, L.C. (ed.): The Development of Social Network Analysis. Adaptive Computation and Machine Learning. Empirical Press, Vancouver (2004)
31. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA **99**(12), 7821–7826 (2002)
32. Godsil, C.D., Royle, G.: Algebraic Graph Theory. Graduate Texts in Mathematics. Springer, Berlin (2001)
33. Gori, M., Maggini, M., Sarti, L.: Exact and approximate graph matching using random walks. IEEE Trans. Pattern Anal. Mach. Intell. **27**(7), 167–256 (2005)
34. Grady, L.: Random walks for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **28**(11), 1768–1783 (2006)
35. Gross, J., Yellen, J.: Graph Theory and Its Applications. CRC Press Inc., Boca Raton, FL (1999)
36. Hethcote, H.W.: The mathematics of infectious diseases. SIAM Rev. **42**(4), 599–653 (2000)
37. Jiang, D., Wang, J.: On-line learning of dynamical systems in the presence of model mismatch and disturbances. IEEE Trans. Neural Netw. **11**(6), 1272–1283 (2000)
38. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1), 39–43 (1953)
39. Kim, H.J., Kim, J.M.: Cyclic topology in complex networks. Phys. Rev. E **72**(3), 036109+ (2005)
40. Kinouchi, O., Martinez, A.S., Lima, G.F., Lourenço, G.M., Risau-Gusman, S.: Deterministic walks in random networks: an application to thesaurus graphs. Physica A **315**, 665–676 (2002)
41. Koschützki, D., Lehmann, K.A., Peeters, L., Richter, S., Tenfelde-Podehl, D., Zlotowski, O.: Centrality indices. In: Brandes, U., Erlebach, T. (eds.) Network Analysis: Methodological Foundations. Lecture Notes in Computer Science, vol. 3418, pp. 16–61. Springer, Berlin (2005)
42. Krugman, P.: The Self-Organizing Economy. Oxford University Press, Oxford (1996)
43. Kuperman, M., Abramson, G.: Small world effect in an epidemiological model. Phys. Rev. Lett. **86**(13), 2909–2912 (2001)
44. Latapy, M., Magnien, C., Vecchio, N.D.: Basic notions for the analysis of large two-mode networks. Soc.Netw. **30**(1), 31–48 (2008)
45. Laumann, E.O., Pappi, F.U.: Networks of Collective Action: A Perspective on Community Influence Systems. Academic Press, New York, NY (1976)
46. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. Internet Math. **6**(1), 29–123 (2009)
47. Liang, J., Wang, Z., Liu, X.: State estimation for coupled uncertain stochastic networks with missing measurements and time-varying delays: the discrete-time case. IEEE Trans. Neural Netw. **20**(5), 781–793 (2009)

48. Lima, G.F., Martinez, A.S., Kinouchi, O.: Deterministic walks in random media. Phys. Rev. Lett. **87**, 010603 (2001)
49. Madras, N., Slade, G.: The Self-Avoiding Walk. Birkhäuser, Boston (1996)
50. May, R.M., Lloyd, A.L.: Infection dynamics on scale-free networks. Phys. Rev. E **64**, 066112 (2001)
51. McAuley, J.J., Costa, Caetano, T.S.: Rich-club phenomenon across complex network hierarchies. Appl. Phys. Lett. **91**, 084103 (2007)
52. Meyn, S.P., Tweedie, R.L.: Markov Chains and Stochastic Stability. Cambridge University Press, Cambridge (2009)
53. Newman, M.E.J.: Assortative mixing in networks. Phys. Rev. Lett. **89**(20), 208701 (2002)
54. Newman, M.E.J.: Mixing patterns in networks. Phys. Rev. E **67**(2), 026126 (2003)
55. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**(2), 167–256 (2003)
56. Newman, M.E.J.: Analysis of weighted networks. Phys. Rev. E **70**, 056131 (2004)
57. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
58. Newman, M.E.J.: Networks: An Introduction. Oxford University Press, Oxford (2010)
59. Noh, J.D., Rieger, H.: Random walks on complex networks. Phys. Rev. Lett. **92**, 118701 (2004)
60. Opsahl, T., Panzarasa, P.: Clustering in weighted networks. Soc. Netw. **31**(2), 155–163 (2009)
61. Opsahl, T., Colizza, V., Panzarasa, P., Ramasco, J.J.: Prominence and control: The weighted rich-club effect. Phys. Rev. Lett. **101**, 168702 (2008)
62. Pastor-Satorras, R., Castellano, C., Mieghem, P.V., Vespignani, A.: Epidemic processes in complex networks. Rev. Mod. Phys. **87**, 925 (2015)
63. Pearson, K.: The problem of the random walk. Nature **72**(1867), 294 (1905)
64. Perra, N., Fortunato, S.: Spectral centrality measures in complex networks. Phys. Rev. E **78**, 036107 (2008)
65. Piraveenan, M., Prokopenko, M., Zomaya, A.Y.: Local assortativeness in scale-free networks. Europhys. Lett. **84**(2), 28002 (2008)
66. Piraveenan, M., Prokopenko, M., Zomaya, A.: Local assortativity and growth of Internet. Eur. Phys. J. B **70**, 275–285 (2009)
67. Restrepo, J.G., Ott, E., Hunt, B.R.: Characterizing the dynamical importance of network nodes and links. Phys. Rev. Lett. **97**, 094102 (2006)
68. Rombach, M.P., Porter, M.A., Fowler, J.H., Mucha, P.J.: Core-periphery structure in networks. SIAM J. Appl. Math. **74**(1), 167–190 (2014)
69. Silva, T.C., Zhao, L.: Uncovering overlapping cluster structures via stochastic competitive learning. Inf. Sci. **247**, 40–61 (2013)
70. Smith, D.A., White, D.R.: Structure and dynamics of the global economy: network analysis of international trade. Soc. Forces **70**(4), 857–893 (1992)
71. Stanley, H.E., Buldyrev, S.V.: Statistical physics – the salesman and the tourist. Nature **413**, 373–374 (2001)
72. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)
73. Watts, D.J.: Small Worlds: The Dynamics of Networks Between Order and Randomness. Princeton Studies in Complexity. Princeton University Press, Princeton (2003)
74. Watts, D.J., Strogatz, S.H.: Collective dynamics of ‘small-world’ networks. Nature **393**(6684), 440–442 (1998)
75. Yang, J., Leskovec, J.: Overlapping communities explain core-periphery organization of networks. Proc. IEEE **102**(12), 1892–1902 (2014)
76. Yang, H.X., Wang, W.X., Lai, Y.C., Xie, Y.B., Wang, B.H.: Control of epidemic spreading on complex networks by local traffic dynamics. Phys. Rev. E **84**, 045101+ (2011)
77. Yang, H.X., Tang, M., Lai, Y.C.: Traffic-driven epidemic spreading in correlated networks. Phys. Rev. E **91**, 062817 (2015)
78. Zeng, Y., Cao, J., Zhang, S., Guo, S., Xie, L.: Random-walk based approach to detect clone attacks in wireless sensor networks. IEEE J. Sel. Areas Commun. **28**(5), 677–691 (2010)

79. Zhang, H.F., Xie, J.R., Tang, M., Lai, Y.C.: Suppression of epidemic spreading in complex networks by local information based behavioral responses. Chaos **24**(4), 043106 (2014)
80. Zhong, M., Shen, K., Seiferas, J.: The convergence-guaranteed random walk and its applications in peer-to-peer networks. IEEE Trans. Comput. **57**(5), 619–633 (2008)
81. Zhou, H.: Distance, dissimilarity index, and network community structure. Phys. Rev. E **67**(6), 061901 (2003)
82. Zhou, S., Mondragon, R.J.: The rich-club phenomenon in the Internet topology. IEEE Commun. Lett. **8**(3), 180–182 (2004)
83. Zhou, T., Fu, Z.Q., Wang, B.H.: Epidemic dynamics on complex networks. Progress Nat. Sci. **16**(5), 452–457 (2006)

Springer

http://www.springer.com/978-3-319-17289-7

Machine Learning in Complex Networks
Christiano Silva, T.; Zhao, L.

2016, XVIII, 331 p. 87 illus., 80 illus. in color., Hardcover
ISBN: 978-3-319-17289-7